Exploiting Background Knowledge for Relation Extraction

Yee Seng Chan and **Dan Roth**
University of Illinois at Urbana-Champaign
{chanys,danr}@illinois.edu

Abstract

Relation extraction is the task of recognizing semantic relations among entities. Given a particular sentence supervised approaches to Relation Extraction employed feature or kernel functions which usually have a single sentence in their scope. The overall aim of this paper is to propose methods for using knowledge and resources that are external to the target sentence, as a way to improve relation extraction. We demonstrate this by exploiting background knowledge such as relationships among the target relations, as well as by considering how target relations relate to some existing knowledge resources. Our methods are general and we suggest that some of them could be applied to other NLP tasks.

1 Introduction

Relation extraction (RE) is the task of detecting and characterizing semantic relations expressed between entities in text. For instance, given the sentence "Cone, a Kansas City native, was originally signed by the Royals and broke into the majors with the team.", one of the relations we might want to extract is the *employment* relation between the pair of entity mentions "Cone" and "Royals". RE is important for many NLP applications such as building an ontology of entities, biomedical information extraction, and question answering.

Prior work have employed diverse approaches towards resolving the task. One approach is to build supervised RE systems using sentences annotated with entity mentions and predefined target relations. When given a new sentence, the RE system has to detect and disambiguate the presence of any predefined relations that might exist between each of the mention pairs in the sentence. In building these systems, researchers used a wide variety of features (Kambhatla, 2004; Zhou et al., 2005; Jiang and Zhai, 2007). Some of the common features used to analyze the target sentence include the words appearing in the sentence, their part-of-speech (POS) tags, the syntactic parse of the sentence, and the dependency path between the pair of mentions. In a related line of work, researchers have also proposed various kernel functions based on different structured representations (e.g. dependency or syntactic tree parses) of the target sentences (Bunescu and Mooney, 2005; Zhou et al., 2007; Zelenko et al., 2003; Zhang et al., 2006). Additionally, researchers have tried to automatically extract examples for supervised learning from resources such as Wikipedia (Weld et al., 2008) and databases (Mintz et al., 2009), or attempted *open* information extraction (IE) (Banko et al., 2007) to extract all possible relations.

In this work, we focus on supervised RE. In prior work, the feature and kernel functions employed are usually restricted to being defined on the various representations (e.g. lexical or structural) of the target sentences. However, in recognizing relations, humans are not thus constrained and rely on an abundance of implicit *world knowledge* or *background information*. What quantifies as world or background knowledge is rarely explored in the RE literature and we do not attempt to provide complete nor precise definitions in this paper. However, we show that by considering the relationship between our relations of interest, as

Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010), pages 152–160,
Beijing, August 2010

well as how they relate to some existing knowledge resources, we improve the performance of RE. Specifically, the contributions of this paper are the following:

- When our relations of interest are clustered or organized in a hierarchical ontology, we show how to use this information to improve performance. By defining appropriate constraints between the predictions of relations at different levels of the hierarchy, we obtain globally coherent predictions as well as improved performance.
- Coreference is a generic relationship that might exists among entity mentions and we show how to exploit this information by assuming that co-referring mentions have no other interesting relations. We capture this intuition by using coreference information to constraint the predictions of a RE system.
- When characterizing the relationship between a pair of mentions, one can use a large encyclopedia such as Wikipedia to infer more knowledge about the two mentions. In this work, after probabilistically mapping mentions to their respective Wikipedia pages, we check whether the mentions are related. Another generic relationship that might exists between a pair of mentions is whether they have a *parent-child* relation and we use this as additional information.
- The sparsity of features (especially lexical features) is a common problem for supervised systems. In this work, we show that one can make fruitful use of unlabeled data, by using word clusters automatically gathered from unlabeled texts as a way of generalizing the lexical features.
- We combine the various relational predictions and background knowledge through a global inference procedure, which we formalize via an Integer Linear Programming (ILP) framework as a constraint optimization problem (Roth and Yih, 2007). This allows us to easily incorporate various constraints that encode the background knowledge.

Roth and Yih (2004) develop a relation extraction approach that exploits constraints among entity types and the relations allowed among them. We extend this view significantly, within a similar computational framework, to exploit relations among target relations, background information and world knowledge, as a way to improve relation extraction and make globally coherent predictions.

In the rest of this paper, we first describe the features used in our basic RE system in Section 2. We then describe how we make use of background knowledge in Section 3. In Section 4, we show our experimental results and perform analysis in Section 5. In Section 6, we discuss related work, before concluding in Section 7.

2 Relation Extraction System

In this section, we describe the features used in our basic relation extraction (RE) system. Given a pair of mentions m_1 and m_2 occurring within the same sentence, the system predicts whether any of the predefined relation holds between the two mentions. Since relations are usually asymmetric in nature, hence in all of our experiments, unless otherwise stated, we distinguish between the argument ordering of the two mentions. For instance, we consider m_1*:emp-org:*m_2 and m_2*:emp-org:*m_1 to be distinct relation types.

Most of the features used in our system are based on the work in (Zhou et al., 2005). In this paper, we propose some new collocation features inspired by word sense disambiguation (WSD). We give an overview of the features in Table 1. Due to space limitations, we only describe the collocation features and refer the reader to (Zhou et al., 2005) for the rest of the features.

2.1 Collocation Features

Following (Zhou et al., 2005), we use a single word to represent the head word of a mention. Since single words might be ambiguous or polysemous, we incorporate local collocation features which were found to be very useful for WSD. Given the head word hw_m of a mention *m*, the collocation feature $C_{i,j}$ refers to the sequence of tokens in the immediate context of hw_m. The offsets *i* and *j* denote the position (relative to hw_m)

Category	Feature
Lexical	hw of m_1
	hw of m_2
	hw of m_1, m_2
	BOW in m_1
	BOW in m_2
	single word between m_1, m_2
	BOW in between m_1, m_2
	bigrams in between m_1, m_2
	first word in between m_1, m_2
	last word in between m_1, m_2
Collocations	$C_{-1,-1}$, $C_{+1,+1}$
	$C_{-2,-1}$, $C_{-1,+1}$, $C_{+1,+2}$
Structural	m_1-in-m_2
	m_2-in-m_1
	#mentions between m_1, m_2
	any word between m_1, m_2
M-lvl	M-lvl of m_1, m_2
and	m_1, m_2 E-maintype
E-type	m_1, m_2 E-subtype
	m_1, m_2 M-lvl and E-maintype
	m_1, m_2 M-lvl and E-subtype
	m_1, m_2 E-subtype and m_1-in-m_2
	m_1, m_2 E-subtype and m_2-in-m_1
Dependency	path between m_1, m_2
	bag-of dep labels between m_1, m_2
	hw of m_1 and dep-parent
	hw of m_2 and dep-parent

Table 1: Features in the basic RE system. The abbreviations are as follows. hw: head word, M-lvl: mention level, E-type: entity type, dep-parent: the word's parent in the dependency tree.

of the first and last token of the sequence respectively. For instance, $C_{-1,+1}$ denotes a sequence of three tokens, consisting of the single token on the immediate left of hw_m, the token hw_m itself, and the single token on the immediate right of hw_m. For each mention, we extract 5 features: $C_{-1,-1}$, $C_{+1,+1}$, $C_{-2,-1}$, $C_{-1,+1}$, and $C_{+1,+2}$.

3 Using Background Knowledge

Now we describe how we inject additional knowledge into our relation extraction system.

3.1 Hierarchy of Relations

When our relations of interest are arranged in a hierarchical structure, one should leverage this information to learn more accurate relation predictors. For instance, assume that our relations are arranged in a two-level hierarchy and we learn two classifiers, one for disambiguating between the first level *coarse-grained* relations, and another for disambiguating between the second level *fine-grained* relations.

Since there are a lot more fine-grained relation types than coarse-grained relation types, we propose using the coarse-grained predictions which should intuitively be more reliable, to improve the fine-grained predictions. We show how to achieve this through defining appropriate constraints between the coarse-grained and fine-grained relations, which can be enforced through the Constrained Conditional Models framework (aka ILP) (Roth and Yih, 2007; Chang et al., 2008). Due to space limitations, we refer interested readers to the papers for more information on the CCM framework.

By doing this, not only are the predictions of both classifiers coherent with each other (thus obtaining better predictions from both classifiers), but more importantly, we are effectively using the (more reliable) predictions of the coarse-grained classifier to constrain the predictions of the fine-grained classifier. To the best of our knowledge, this approach for RE is novel.

In this paper, we work on the NIST Automatic Content Extraction (ACE) 2004 corpus. ACE defines several coarse-grained relations such as *employment/membership*, *geo-political entity (GPE) affiliation*, etc. Each coarse-grained relation is further refined into several fine-grained relations[1] and each fine-grained relation has a unique parent coarse-grained relation. For instance, the fine-grained relations *employed as ordinary staff*, *employed as an executive*, etc. are children relations of *employment/membership*.

Let m_i and m_j denote a pair of mentions i and j drawn from a document containing N mentions. Let $R_{i,j}$ denote a relation between m_i and m_j, and let $\mathcal{R} = \{R_{i,j}\}$, where $1 \leq i, j \leq N; i \neq j$ denote the set of relations in the document. Also, we denote the set of predefined coarse-grained relation types and fine-grained relation types as $\mathcal{L}_{Rc}$ and $\mathcal{L}_{Rf}$ respectively. Since there could possibly be no relation between a mention pair, we add the *null* label to $\mathcal{L}_{Rc}$ and $\mathcal{L}_{Rf}$, allowing our classifiers to predict *null* for $R_{i,j}$. Finally, for a fine-grained relation type rf, let $\mathcal{V}(rf)$ denote its parent coarse-grained relation type.

[1] With the exception of the *Discourse* coarse-grained relation.

We learn two classifiers, one for disambiguating between the coarse-grained relations and one for disambiguating between the fine-grained relations. Let θ_c and θ_f denote the feature weights learned for predicting coarse-grained and fine-grained relations respectively. Let $p_R(rc) = \log P_c(rc|m_i, m_j; \theta_c)$ be the log probability that relation R is predicted to be of coarse-grained relation type rc. Similarly, let $p_R(rf) = \log P_f(rf|m_i, m_j; \theta_f)$ be the log probability that relation R is predicted to be of fine-grained relation type rf. Let $x_{\langle R,rc\rangle}$ be a binary variable which takes on the value of 1 if relation R is labeled with the coarse-grained label rc. Similarly, let $y_{\langle R,rf\rangle}$ be a binary variable which takes on the value of 1 if relation R is labeled with the fine-grained label rf. Our objective function is then:

$$\max \sum_{R\in\mathcal{R}} \sum_{rc\in\mathcal{L}_{Rc}} p_R(rc)\cdot x_{\langle R,rc\rangle} + \sum_{R\in\mathcal{R}} \sum_{rf\in\mathcal{L}_{Rf}} p_R(rf)\cdot y_{\langle R,rf\rangle} \quad (1)$$

subject to the following constraints:

$$\sum_{rc\in\mathcal{L}_{Rc}} x_{\langle R,rc\rangle} = 1 \quad \forall R\in\mathcal{R} \quad (2)$$

$$\sum_{rf\in\mathcal{L}_{Rf}} y_{\langle R,rf\rangle} = 1 \quad \forall R\in\mathcal{R} \quad (3)$$

$$x_{\langle R,rc\rangle} \in \{0,1\} \quad \forall R\in\mathcal{R}, rc\in\mathcal{L}_{Rc} \quad (4)$$

$$y_{\langle R,rf\rangle} \in \{0,1\} \quad \forall R\in\mathcal{R}, rf\in\mathcal{L}_{Rf} \quad (5)$$

Equations (2) and (3) require that each relation can only be assigned one coarse-grained label and one fine-grained label. Equations (4) and (5) indicate that $x_{\langle R,rc\rangle}$ and $y_{\langle R,rf\rangle}$ are binary variables. Two more constraints follow:

$$x_{\langle R,rc\rangle} \leq \sum_{\{rf\in\mathcal{L}_{Rf}|\mathcal{V}(rf)=rc\}} y_{\langle R,rf\rangle} \quad \forall R\in\mathcal{R}, rc\in\mathcal{L}_{Rc} \quad (6)$$

$$y_{\langle R,rf\rangle} \leq x_{\langle R,\mathcal{V}(rf)\rangle} \quad \forall R\in\mathcal{R}, rf\in\mathcal{L}_{Rf} \quad (7)$$

The logical form of Equation (6) can be written as: $x_{\langle R,rc\rangle} \Rightarrow y_{\langle R,rf_1\rangle} \vee y_{\langle R,rf_2\rangle} \ldots \vee y_{\langle R,rf_n\rangle}$, where $rf_1, rf_2, \ldots, rf_n$ are (child) fine-grained relations of the coarse-grained relation rc. This states that if we assign rc to relation R, then we must also assign to R a fine-grained relation rf which is a *child* of rc. The logical form of Equation (7) can be written as: $y_{\langle R,rf\rangle} \Rightarrow x_{\langle R,\mathcal{V}(rf)\rangle}$. This captures the inverse relation and states that if we assign rf to R, then we must also assign to R the relation type $\mathcal{V}(rf)$, which is the *parent* of rf. Together, Equations (6) and (7) constrain the predictions of the coarse-grained and fine-grained classifiers to be coherent with each other. Finally, we note that one could automatically translate logical constraints into linear inequalities (Chang et al., 2008).

This method is general and is applicable to other NLP tasks where a hierarchy exists, such as WSD and question answering. For instance, in WSD, one can predict coarse-grained and fine-grained senses using suitably defined sense inventories and then perform inference via ILP to obtain coherent predictions.

art:	E_i ∈{gpe, org, per}, E_j ∈{fac, gpe, veh, wea}
emp-org:	E_i ∈{gpe, org, per}, E_j ∈{gpe, org, per}
gpe-aff:	E_i ∈{gpe, org, per}, E_j ∈{gpe, loc}
other-aff:	E_i ∈{gpe, org, per}, E_j ∈{gpe, loc}
per-soc:	E_i ∈{per}, E_j ∈{per}

Table 2: Entity type constraints.

3.2 Entity Type Constraints

Each mention in ACE-2004 is annotated with one of seven coarse-grained entity types: person (per), organization (org), location (loc), geo-political entity (gpe), facility (fac), vehicle (veh), and weapon (wea).

Roth and Yih (2007) had shown that entity type information is useful for constraining the possible labels that a relation R can assume. For instance, both mentions involved in a *personal/social* relation must be of entity type *per*. In this work, we gather such information from the ACE-2004 documentation and inject it as constraints (on the coarse-grained relations) into our system. Due to space limitations, we do not state the constraint equations or objective function here, but we list the entity type constraints we imposed for each coarse-grained relation m_i-R-m_j in Table

2^2, where E_i (E_j) denotes the allowed set of entity types for mention m_i (m_j). Applying the entity type information improves the predictions of the coarse-grained classifier and this in turn could improve the predictions of the fine-grained classifier.

3.3 Using Coreference Information

We can also utilize the coreference relations among entity mentions. Assuming that we know mentions m_i and m_j are coreferent with each other, then there should be no relation between them[3]. Let $z_{\langle i,j \rangle}$ be a binary variable which takes on the value of 1 if mentions m_i and m_j are coreferent, and 0 if they are not. When $z_{\langle i,j \rangle}$=1, we capture the above intuition with the following constraints:

$$z_{\langle i,j \rangle} \leq x_{\langle R_{i,j},null \rangle} \quad (8)$$

$$z_{\langle i,j \rangle} \leq y_{\langle R_{i,j},null \rangle} \quad (9)$$

which can be written in logical form as: $z_{\langle i,j \rangle} \Rightarrow x_{\langle R_{i,j},null \rangle}$, and $z_{\langle i,j \rangle} \Rightarrow y_{\langle R_{i,j},null \rangle}$. We add the following to our objective function in Equation (1):

$$\sum_{m_i,m_j \in \mathbf{m}^2} co_{\langle i,j \rangle} \cdot z_{\langle i,j \rangle} + \bar{co}_{\langle i,j \rangle} \cdot (1 - z_{\langle i,j \rangle}) \quad (10)$$

where $\mathbf{m}$ is the set of mentions in a document, $co_{\langle i,j \rangle}$ and $\bar{co}_{\langle i,j \rangle}$ are the log probabilities of predicting that m_i and m_j are coreferent and not coreferent respectively. In this work, we assume we are given coreference information, which is available from the ACE annotation.

3.4 Using Knowledge from Wikipedia

We propose two ways of using Wikipedia to gather features for relation extraction. Wikipedia is a huge online encyclopedia and mainly contains articles describing entities or concepts.

The first intuition is that if we are able to correctly map a pair of mentions m_i and m_j to their corresponding Wikipedia article (assuming they are represented in Wikipedia), we could use the content on their Wikipedia pages to check whether they are related.

In this work, we use a *Wiki* system (Ratinov et al., 2010) which performs context-sensitive mapping of mentions to Wikipedia pages. In their work, the authors first identify phrases or mentions that could be mapped. The correct Wikipedia article for each mention is then probabilistically predicted using a combination of features based on Wikipedia hyperlink structure, semantic coherence, etc. The authors' own evaluation results indicate that the performance of their system ranges from 70–80%. When given a pair of mentions and the system returns the Wikipedia page for either one of the mentions, we introduce a feature:

$$w_1(m_i, m_j) = \begin{cases} 1, & \text{if } A_{m_i}(m_j) \\ & \text{or } A_{m_j}(m_i) \\ 0, & \text{otherwise} \end{cases}$$

where $A_{m_i}(m_j)$ returns true if the head extent of m_j is found (via simple string matching) in the predicted Wikipedia article of m_i. The interpretation of $A_{m_j}(m_i)$ is similar. We introduce a new feature into the RE system by combining $w_1(m_i, m_j)$ with m_i, m_j E-maintype (defined as in Table 1).

The second feature based on Wikipedia is as follows. It will be useful to check whether there is any *parent-child* relationship between two mentions. Intuitively, this will be useful for recognizing several relations such as *physical part-whole* (e.g. a city is part of a state), *subsidiary* (a company is a child-company of another), *citizenship* (a person is a citizen of a country), etc.

Given a pair of mentions m_i and m_j, we use a *Parent-Child* system (Do and Roth, 2010) to predict whether they have a *parent-child* relation. To achieve this, the system first gathers all Wikipedia articles that are related to m_i and m_j. It then uses the words in these pages and the category ontology of Wikipedia to make its *parent-child* predictions, while respecting certain defined constraints. In this work, we use its prediction as follows:

$$w_2(m_i, m_j) = \begin{cases} 1, & \text{if } \textit{parent-child}(m_i, m_j) \\ 0, & \text{otherwise} \end{cases}$$

[2]We do not impose entity type constraints on the coarse-grained relations *disc* and *phys*.

[3]In this work, we assume that no relations are reflexive. After the experiments in this paper are performed, we verified that in the ACE corpus we used, less than 1% of the relations are reflexive.

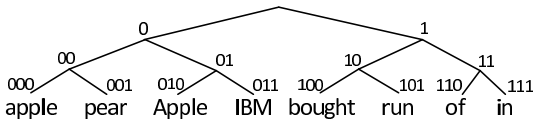

Figure 1: An example of Brown word cluster hierarchy from (Koo et al., 2008).

where we combine $w_2(m_i, m_j)$ with m_i, m_j E-maintype, introducing this as a new feature into our RE system.

3.5 Using Word Clusters

An inherent problem faced by supervised systems is that of data sparseness. To mitigate such issues in the lexical features, we use word clusters which are automatically generated from unlabeled texts. In this work, we use the Brown clustering algorithm (Brown et al., 1992), which has been shown to improve performance in various NLP applications such as dependency parsing (Koo et al., 2008), named entity recognition (Ratinov and Roth, 2009), and relation extraction (Boschee et al., 2005). The algorithm performs a hierarchical clustering of the words and represents them as a binary tree.

Each word is uniquely identified by its path from the root and every path is represented with a bit string. Figure 1 shows an example clustering where the maximum path length is 3. By using path prefixes of different lengths, one can obtain clusterings at different granularity. For instance, using prefixes of length 2 will put *apple* and *pear* into the same cluster, *Apple* and *IBM* into the same cluster, etc. In our work, we use clusters generated from New York Times text and simply use a path prefix of length 10. When Brown clusters are used in our system, all lexical features consisting of single words will be duplicated. For instance, for the feature *hw of m1*, one new feature which is the length-10 bit string path representing the original lexical head word of m1, will be introduced and presented to the classifier as a string feature.

4 Experiments

We used the ACE-2004 dataset (catalog LDC2005T09 from the Linguistic Data Consortium) to conduct our experiments. ACE-2004 defines 7 coarse-grained relations and 23 fine-grained relations. In all of our experiments, unless otherwise stated, we explicitly model the argument order (of the mentions) when asked to disambiguate the relation between a pair of mentions. Hence, we built our coarse-grained classifier with 15 relation labels to disambiguate between (two for each coarse-grained relation type and a *null* label when the two mentions are not related). Likewise, our fine-grained classifier has to disambiguate between 47 relation labels. In the dataset, relations do not cross sentence boundaries.

For our experiments, we trained regularized averaged perceptrons (Freund and Schapire, 1999), implemented within the Sparse Network of Winnow framework (Carlson et al., 1999), one for predicting the coarse-grained relations and another for predicting the fine-grained relations. Since the dataset has no split of training, development, and test sets, we followed prior work (Jiang and Zhai, 2007) and performed 5-fold cross validation to obtain our performance results. For simplicity, we used 5 rounds of training and a regularization parameter of 1.5 for the perceptrons in all our experiments. Finally, we concentrate on the evaluation of fine-grained relations.

4.1 Performance of the Basic RE system

As a gauge on the performance of our basic relation extraction system BasicRE using only the features described in Section 2, we compare against the state-of-the-art feature-based RE system of Jiang and Zhai (2007). However, we note that in that work, the authors performed their evaluation using *undirected* coarse-grained relations. That is, they do not distinguish on argument order of mentions and the classifier has to decide among 8 relation labels (7 coarse-grained relation types and a *null* label). Performing 5-fold cross validation on the news wire (nwire) and broadcast news (bnews) corpora in the ACE-2004 dataset, they reported a F-measure of 71.5 using a maximum entropy classifier[4]. Evaluating BasicRE on the same setting,

[4] After they heuristically performed feature selection and applied the heuristics giving the best evaluation performance, they obtained a result of 72.9.

Features	All nwire			10% of nwire		
	Rec%	Pre%	F1%	Rec%	Pre%	F1%
BasicRE	49.9	51.0	50.5	33.2	29.0	31.0
+Hier	+1.3	+1.3	+1.3	+1.1	+1.2	+1.1
+Hier+relEntC	+1.5	+2.0	+1.8	+3.3	+3.5	+3.4
+Coref	$\sim$	+1.4	+0.7	−0.1	+1.0	+0.5
+Wiki	+0.2	+1.9	+1.0	+1.5	+2.5	+2.0
+Cluster	−0.2	+3.2	+1.4	−0.7	+3.9	+1.7
+ALL	+1.5	+6.7	+3.9	+4.7	+10.2	+7.6

Table 3: BasicRE gives the performance of our basic RE system on predicting fine-grained relations, obtained by performing 5-fold cross validation on only the news wire corpus of ACE-2004. Each subsequent row +Hier, +Hier+relEntC, +Coref, +Wiki, and +Cluster gives the *individual* contribution from using each knowledge. The bottom row +ALL gives the performance improvements from adding +Hier+relEntC+Coref+Wiki+Cluster. $\sim$ indicates no change in score.

we obtained a competitive F-measure of 71.2[5].

4.2 Experimental Settings for Evaluating Fine-grained Relations

Two of our knowledge sources, the Wiki system described in Section 3.4 and the word clusters described in Section 3.5, assume inputs of mixed-cased text. We note that the bnews corpus of ACE-2004 is entirely in lower-cased text. Hence, we use only the nwire corpus for our experiments here, from which we gathered 28,943 relation instances and 2,226 of those have a valid (non-null) relation[6].

We also propose the following experimental setting. First, since we made use of coreference information, we made sure that while performing our experiments, all instances from the same document are either all used as training data or all used as test data. Prior work in RE had not ensured this, but we argue that this provides a more realistic setting. Our own experiments indicate that this results in a 1-2% lower performance on fine-grained relations.

Secondly, prior work calculate their performance on relation extraction at the level of *mentions*. That is, each mention pair extracted is scored individually. An issue with this way of scoring on the ACE corpus is that ACE annotators rarely duplicate a relation link for coreferent mentions. For instance, assume that mentions m_i, m_j, and m_k exist in a given sentence, mentions m_i and m_j are coreferent, and the annotator establishes a particular relation type r between m_j and m_k. The annotator will not usually duplicate the same relation r between m_i and m_k and thus the label between these two mentions is then *null*. We are not suggesting that this is an incorrect approach, but clearly there is an issue since an important goal of performing RE is to populate or build an ontology of entities and establish the relations existing among the entities. Thus, we evaluate our performance at the *entity-level*.[7] That is, given a pair of entities, we establish the set of relation types existing between them, based on their mention annotations. Then we calculate recall and precision based on these established relations. Of course, performing such an evaluation requires knowledge about the coreference relations and in this work, we assume we are given this information.

4.3 Knowledge-Enriched System

Evaluating our system BasicRE (trained only on the features described in Section 2) on the nwire corpus, we obtained a F1 score of 50.5, as shown in Table 3. Next, we exploited the relation hierarchy as in Section 3.1 and obtained an improvement of 1.3, as shown in the row +Hier. Next, we added the entity type constraints of Section

[5]Using 10 rounds of training and a regularization parameter of 2.5 improves the result to 72.2. In general, we found that more rounds of training and a higher regularization value benefits coarse-grained relation classification, but not fine-grained relation classification.

[6]The number of relation instances in the nwire and bnews corpora are about the same.

[7]Our experiments indicate that performing the usual evaluation on mentions gives similar performance figures and the trend in Table 3 stays the same.

3.2. Remember that these constraints are imposed on the coarse-grained relations. Thus, they would only affect the fine-grained relation predictions if we also exploit the relation hierarchy. In the table, we show that all the background knowledge helped to improve performance, providing a total improvement of 3.9 to our basic RE system. Though the focus of this work is on fine-grained relations, our approach also improves the performance of coarse-grained relation predictions. BasicRE obtains a F1 score of 65.3 on coarse-grained relations and exploiting background knowledge gives a total improvement of 2.9.

5 Analysis

We explore the situation where we have very little training data. We assume during each cross validation fold, we are given only 10% of the training data we originally had. Previously, when performing 5-fold cross validation on 2,226 valid relation instances, we had about 1780 as training instances in each fold. Now, we assume we are only given about 178 training instances in each fold. Under this condition, BasicRE gives a F1 score of 31.0 on fine-grained relations. Adding all the background knowledge gives an improvement of 7.6 and this represents an error reduction of 39% when measured against the performance difference of 50.5 (31.0) when we have 1780 training instances vs. 178 training instances. On the coarse-grained relations, BasicRE gives a F1 score of 51.1 and exploiting background knowledge gives a total improvement of 5.0.

We also tabulated the list of fine-grained relations that improved by more than 1 F1 score when we incorporated +Wiki, on the experiment using all of nwire data: *phys:near* (physically near), *other-aff:ideology* (ideology affiliation), *art:user-or-owner* (user or owner of artifact), *per-soc:business* (business relationship), *phys:part-whole* (physical part-whole), *emp-org:subsidiary* (organization subsidiary), and *gpe-aff:citizen-or-resident* (citizen or resident). Most of these intuitively seemed to be information one would find being mentioned in an encyclopedia.

6 Related Work

Few prior work has explored using background knowledge to improve relation extraction performance. Zhou et al. (2008) took advantage of the hierarchical ontology of relations by proposing methods customized for the perceptron learning algorithm and support vector machines. In contrast, we propose a generic way of using the relation hierarchy which at the same time, gives globally coherent predictions and allows for easy injection of knowledge as constraints. Recently, Jiang (2009) proposed using features which are common across all relations. Her method is complementary to our approach, as she does not consider information such as the relatedness between different relations. On using semantic resources, Zhou et al. (2005) gathered two gazettes, one containing country names and another containing words indicating personal relationships. In relating the tasks of RE and coreference resolution, Ji et al. (2005) used the output of a RE system to rescore coreference hypotheses. In our work, we reverse the setting and explore using coreference to improve RE.

7 Conclusion

In this paper, we proposed a broad range of methods to inject background knowledge into a relation extraction system. Some of these methods, such as exploiting the relation hierarchy, are general in nature and could be easily applied to other NLP tasks. To combine the various relation predictions and knowledge, we perform global inference within an ILP framework. Besides allowing for easy injection of knowledge as constraints, this ensures globally coherent models and predictions.

Acknowledgements This research was partly sponsored by Air Force Research Laboratory (AFRL) under prime contract no. FA8750-09-C-0181. We thank Ming-Wei Chang and James Clarke for discussions on this research.

References

Banko, Michele, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007.

Open information extraction from the web. In *Proceedings of IJCAI-07*, pages 2670–2676.

Boschee, Elizabeth, Ralph Weischedel, and Alex Zamanian. 2005. Automatic information extraction. In *Proceedings of the International Conference on Intelligence Analysis*.

Brown, Peter F., Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Bunescu, Razvan C. and Raymond J. Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of HLT/EMNLP-05*, pages 724–731.

Carlson, Andrew J., Chad M. Cumby, Jeff L. Rosen, and Dan Roth. 1999. The SNoW learning architecture. Technical Report UIUCDCS-R-99-2101, UIUC Computer Science Department, May.

Chang, Ming-Wei, Lev Ratinov, Nicholas Rizzolo, and Dan Roth. 2008. Learning and inference with constraints. In *Proceedings of AAAI-08*, pages 1513–1518.

Do, Quang and Dan Roth. 2010. On-the-fly constraint-based taxonomic relation identification. Technical report, University of Illinois. http://L2R.cs.uiuc.edu/∼danr/Papers/DoRo10.pdf.

Freund, Yoav and Robert E. Schapire. 1999. Large margin classification using the perceptron algorithm. *Machine Learning*, 37(3):277–296.

Ji, Heng, David Westbrook, and Ralph Grishman. 2005. Using semantic relations to refine coreference decisions. In *Proceedings of HLT/EMNLP-05*, pages 17–24.

Jiang, Jing and ChengXiang Zhai. 2007. A systematic exploration of the feature space for relation extraction. In *Proceedings of HLT-NAACL-07*, pages 113–120.

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

Kambhatla, Nanda. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proceedings of ACL-04*, pages 178–181.

Koo, Terry, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL-08:HLT*, pages 595–603.

Mintz, Mike, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL-IJCNLP-09*, pages 1003–1011.

Ratinov, Lev and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of CoNLL-09*, pages 147–155.

Ratinov, Lev, Doug Downey, and Dan Roth. 2010. Wikification for information retrieval. Technical report, University of Illinois. http://L2R.cs.uiuc.edu/∼danr/Papers/RatinovDoRo10.pdf.

Roth, Dan and Wen Tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of CoNLL-04*, pages 1–8.

Roth, Dan and Wen Tau Yih. 2007. Global inference for entity and relation identification via a linear programming formulation. In Getoor, Lise and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press.

Weld, Daniel S., Raphael Hoffman, and Fei Wu. 2008. Using wikipedia to bootstrap open information extraction. *ACM SIGMOD Special Issue on Managing Information Extraction*, 37(4):62–68.

Zelenko, Dmitry, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106.

Zhang, Min, 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 COLING-ACL-06*, pages 825–832.

Zhou, Guodong, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of ACL-05*.

Zhou, GuoDong, Min Zhang, DongHong Ji, and QiaoMing Zhu. 2007. Tree kernel-based relation extraction with context-sensitive structured parse tree information. In *Proceedings of EMNLP-CoNLL-07*, pages 728–736.

Zhou, Guodong, Min Zhang, Dong-Hong Ji, and Qiaoming Zhu. 2008. Hierarchical learning strategy in semantic relation extraction. *Information Processing & Management*, 44(3):1008–1021.