Exploiting Syntactico-Semantic Structures for Relation Extraction

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

Abstract

In this paper, we observe that there exists a second dimension to the relation extraction (RE) problem that is orthogonal to the relation type dimension. We show that most of these second dimensional structures are relatively constrained and not difficult to identify. We propose a novel algorithmic approach to RE that starts by first identifying these structures and then, within these, identifying the semantic type of the relation. In the real RE problem where relation arguments need to be identified, exploiting these structures also allows reducing pipelined propagated errors. We show that this RE framework provides significant improvement in RE performance.

1 Introduction

Relation extraction (RE) has been defined as the task of identifying a given set of semantic binary relations in text. For instance, given the span of text "... the Seattle zoo ...", one would like to extract the relation that "the Seattle zoo" is located-at "Seattle". RE has been frequently studied over the last few years as a supervised learning task, learning from spans of text that are annotated with a set of semantic relations of interest. However, most approaches to RE have assumed that the relations' arguments are given as input (Chan and Roth, 2010; Jiang and Zhai, 2007; Jiang, 2009; Zhou et al., 2005), and therefore offer only a partial solution to the problem.

Conceptually, this is a rather simple approach as *all* spans of texts are treated uniformly and are being mapped to one of several relation types of interest. However, these approaches to RE require a large amount of manually annotated training data to achieve good performance, making it difficult to expand the set of target relations. Moreover, as we show, these approaches become brittle when the relations' arguments are not given but rather need to be identified in the data too.

In this paper we build on the observation that there exists a second dimension to the relation extraction problem that is orthogonal to the *relation type* dimension: all relation types are expressed in one of several constrained syntactico-semantic structures. As we show, identifying where the text span is on the *syntactico-semantic structure* dimension first, can be leveraged in the RE process to yield improved performance. Moreover, working in the second dimension provides robustness to the *real* RE problem, that of identifying arguments along with the relations between them.

For example, in "the Seattle zoo", the entity mention "Seattle" modifies the noun "zoo". Thus, the two mentions "Seattle" and "the Seattle zoo", are involved in what we later call a *premodifier relation*, one of several syntactico-semantic structures we identify in Section 3.

We highlight that all relation types can be expressed in one of several syntactico-semantic structures – Premodifiers, Possessive, Preposition, Formulaic and Verbal. As it turns out, most of these structures are relatively constrained and are not difficult to identify. This suggests a novel algorithmic approach to RE that starts by first identifying these structures and then, within these, identifying the semantic type of the relation. Not only does this approach provide significantly improved RE perfor-

mance, it carries with it two additional advantages.

First, leveraging the syntactico-semantic structure is especially beneficial in the presence of small amounts of data. Second, and more important, is the fact that exploiting the syntactico-semantic dimension provides several new options for dealing with the *full* RE problem – incorporating the argument identification into the problem. We explore one of these possibilities, making use of the constrained structures as a way to aid in the identification of the relations' arguments. We show that this already provides significant gain, and discuss other possibilities that can be explored. The contributions of this paper are summarized below:

- We highlight that all relation types are expressed as one of several syntactico-semantic structures and show that most of these are relatively constrained and not difficult to identify. Consequently, working first in this structural dimension can be leveraged in the RE process to improve performance.

- We show that when one does not have a large number of training examples, exploiting the syntactico-semantic structures is crucial for RE performance.

- We show how to leverage these constrained structures to improve RE when the relations' arguments are not given. The constrained structures allow us to jointly entertain argument candidates and relations built with them as arguments. Specifically, we show that considering argument candidates which otherwise would have been discarded (provided they exist in syntactico-semantic structures), we reduce error propagation along a standard pipeline RE architecture, and that this joint inference process leads to improved RE performance.

In the next section, we describe our relation extraction framework that leverages the syntactico-semantic structures. We then present these structures in Section 3. We describe our mention entity typing system in Section 4 and features for the RE system in Section 5. We present our RE experiments in Section 6 and perform analysis in Section 7, before concluding in Section 8.

$\mathcal{S}$ = {premodifier, possessive, preposition, formulaic}
gold mentions in training data $\mathcal{M}_{train}$
$\mathcal{D}_g = \{(m_i, m_j) \in \mathcal{M}_{train} \times \mathcal{M}_{train} \mid$
  m_i in same sentence as $m_j \wedge i \neq j \wedge i < j\}$
RE_{base} = RE classifier trained on $\mathcal{D}_g$

$D_s = \emptyset$
for each $(m_i, m_j) \in \mathcal{D}_g$
do
  p = structure inference on (m_i, m_j) using patterns
  if $p \in \mathcal{S} \vee (m_i, m_j)$ was annotated with a $\mathcal{S}$ structure
    $\mathcal{D}_s = \mathcal{D}_s \cup (m_i, m_j)$
done
RE_s = RE classifier trained on $\mathcal{D}_s$

Output: RE_{base} and RE_s

Figure 1: Training a regular baseline RE classifier RE_{base} and a RE classifier leveraging syntactico-semantic structures RE_s.

2 Relation Extraction Framework

In Figure 1, we show the algorithm for training a typical baseline RE classifier (RE_{base}), and for training a RE classifier that leverages the syntactico-semantic structures (RE_s).

During evaluation and when the *gold* mentions are already annotated, we apply RE_s as follows. When given a test example mention pair (x_i,x_j), we perform structure inference on it using the patterns described in Section 3. If (x_i,x_j) is identified as having any of the four syntactico-semantic structures $\mathcal{S}$, apply RE_s to predict the relation label, else apply RE_{base}.

Next, we show in Figure 2 our joint inference algorithmic framework that leverages the syntactico-semantic structures for RE, when mentions need to be *predicted*. Since the structures are fairly constrained, we can use them to consider mention candidates that are originally predicted as non mentions. As shown in Figure 2, we conservatively include such mentions when forming mention pairs, provided their null labels are predicted with a low probability t[1].

[1]In this work, we arbitrary set t=0.2. After the experiments, and in our own analysis, we observe that t=0.25 achieves better performance. Besides using the probability of the 1-best prediction, one could also for instance, use the probability difference between the first and second best predictions. However, selecting an optimal t value is not the main focus of this work.

$\mathcal{S}$ = {premodifier, possessive, preposition, formulaic}
candidate mentions $\mathcal{M}_{cand}$
Let $L_m = \underset{y}{\text{argmax}} P_{MET}(y|m,\theta), m \in \mathcal{M}_{cand}$
selected mentions $\mathcal{M}_{sel} = \{m \in \mathcal{M}_{cand} \mid$
$L_m \neq null \vee P_{MET}(null|m,\theta) \leq t\}$
$\mathcal{Q}_{hasNull} = \{(m_i, m_j) \in \mathcal{M}_{sel} \times \mathcal{M}_{sel} \mid$
m_i in same sentence as $m_j \wedge i \neq j \wedge i < j \wedge$
$(L_{m_i} \neq null \vee L_{m_j} \neq null)\}$
Let pool of relation predictions $\mathcal{R} = \emptyset$

for each $(m_i, m_j) \in \mathcal{Q}_{hasNull}$
do
p = structure inference on (m_i, m_j) using patterns
if $p \in \mathcal{S}$
r = relation prediction for (m_i, m_j) using RE_s
$\mathcal{R} = \mathcal{R} \cup r$
else if $L_{m_i} \neq null \wedge L_{m_j} \neq null$
r = relation prediction for (m_i, m_j) using RE_{base}
$\mathcal{R} = \mathcal{R} \cup r$
done

Output: $\mathcal{R}$

Figure 2: RE using predicted mentions and patterns. Abbreviations: L_m: predicted entity label for mention m using the mention entity typing (MET) classifier described in Section 4; P_{MET}: prediction probability according to the MET classifier; t: used for thresholding.

There is a large body of work in using patterns to extract relations (Fundel et al., 2007; Greenwood and Stevenson, 2006; Zhu et al., 2009). However, these works operate along the first dimension, that of using patterns to mine for *relation type* examples. In contrast, in our RE framework, we apply patterns to identify the syntactico-semantic structure dimension first, and leverage this in the RE process. In (Roth and Yih, 2007), the authors used entity types to constrain the (first dimensional) relation types allowed among them. In our work, although a few of our patterns involve semantic type comparison, most of the patterns are syntactic in nature.

In this work, we performed RE evaluation on the NIST Automatic Content Extraction (ACE) corpus. Most prior RE evaluation on ACE data assumed that mentions are already pre-annotated and given as input (Chan and Roth, 2010; Jiang and Zhai, 2007; Zhou et al., 2005). An exception is the work of (Kambhatla, 2004), where the author evaluated on the ACE-2003 corpus. In that work, the author did not address the pipelined errors propagated from the mention identification process.

3 Syntactico-Semantic Structures

In this paper, we performed RE on the ACE-2004 corpus. In ACE-2004 when the annotators tagged a pair of mentions with a relation, they also specified the type of syntactico-semantic structure[2]. ACE-2004 identified five types of structures: premodifier, possessive, preposition, formulaic, and verbal. We are unaware of any previous computational approaches that recognize these structures automatically in text, as we do, and use it in the context of RE (or any other problem). In (Qian et al., 2008), the authors reported the recall scores of their RE system on the various syntactico-semantic structures. But they do not attempt to recognize nor leverage these structures.

In this work, we focus on detecting the first four structures. These four structures cover 80% of the mention pairs having valid semantic relations (we give the detailed breakdown in Section 7) and we show that they are relatively easy to identify using simple rules or patterns. In this section, we indicate mentions using square bracket pairs, and use m_i and m_j to represent a mention pair. We now describe the four structures.

Premodifier relations specify the proper adjective or proper noun premodifier and the following noun it modifies, e.g.: [the [Seattle] zoo]

Possessive indicates that the first mention is in a possessive case, e.g.: [[California] 's Governor]

Preposition indicates that the two mentions are semantically related via the existence of a preposition, e.g.: [officials] in [California]

Formulaic The ACE04 annotation guideline[3] indicates the annotation of several formulaic relations, including for example address: [Medford] , [Massachusetts]

[2] ACE-2004 termed it as lexical condition. We use the term syntactico-semantic structure in this paper as the mention pair exists in specific syntactic structures, and we use rules or patterns that are syntactically and semantically motivated to detect these structures.

[3] http://projects.ldc.upenn.edu/ace/docs/EnglishRDCV4-3-2.PDF

Structure type	Pattern
Premodifier	Basic pattern: [u* [v+] w+] , where u, v, w represent words Each w is a noun or adjective If u* is not empty, then u*: JJ+ ∨ JJ "and" JJ? ∨ CD JJ* ∨ RB DT JJ? ∨ RB CD JJ ∨ DT (RB|JJ|VBG|VBD|VBN|CD)? Let w_1 = first word in w+. $w_1 \neq$ "'s" and POS tag of $w_1 \neq$ POS Let v_l = last word in v+. POS tag of $v_l \neq$ PRP$ nor WP$
Possessive	Basic pattern: [u? [v+] w+] , where u, v, w represent words Let w_1 = first word in w+. If w_1 = "'s" ∨ POS tag of w_1 = POS, accept mention pair Let v_l = last word in v+. If POS tag of v_l = PRP$ or WP$, accept mention pair
Preposition	Basic pattern: [m_i] v* [m_j], where v represent words and number of prepositions in the text span v* between them = 0, 1, or 2 If satisfy pattern: IN [m_i][m_j], accept mention pair If satisfy pattern: [m_i] (IN|TO) [m_j], accept mention pair If all labels in $\mathcal{L}_d$ start with "prep", accept mention pair
Formulaic	If satisfy pattern: [m_i] / [m_j] ∧ $E_c(m_i)$ = PER ∧ $E_c(m_j)$ = ORG, accept mention pair If satisfy pattern: [m_i][m_j] If $E_c(m_i)$ = PER ∧ $E_c(m_j)$ = ORG ∨ GPE, accept mention pair

Table 1: Rules and patterns for the four syntactico-semantic structures. Regular expression notations: '*' matches the preceding element zero or more times; '+' matches the preceding element one or more times; '?' indicates that the preceding element is optional; '|' indicates or. Abbreviations: $E_c(m)$: coarse-grained entity type of mention m; $\mathcal{L}_d$: labels in dependency path between the headword of two mentions. We use square brackets '[' and ']' to denote mention boundaries. The '/' in the *Formulaic* row denotes the occurrence of a lexical '/' in text.

In this rest of this section, we present the rules/patterns for detecting the above four syntactico-semantic structure, giving an overview of them in Table 1. We plan to release all of the rules/patterns along with associated code[4]. Notice that the patterns are intuitive and mostly syntactic in nature.

3.1 Premodifier Structures

- We require that one of the mentions completely include the other mention. Thus, the basic pattern is [u* [v+] w+].

- If u* is not empty, we require that it satisfies any of the following POS tag sequences: JJ+ ∨ JJ and JJ? ∨ CD JJ*, etc. These are (optional) POS tag sequences that normally start a valid noun phrase.

- We use two patterns to differentiate between premodifier relations and possessive relations, by checking for the existence of POS tags PRP$, WP$, POS, and the word "'s".

[4]http://cogcomp.cs.illinois.edu/page/publications

3.2 Possessive Structures

- The basic pattern for possessive is similar to that for premodifier: [u? [v+] w+]

- If the word immediately following v+ is "'s" or its POS tag is "POS", we accept the mention pair. If the POS tag of the last word in v+ is either PRP$ or WP$, we accept the mention pair.

3.3 Preposition Structures

- We first require the two mentions to be non-overlapping, and check for the existence of patterns such as "IN [m_i] [m_j]" and "[m_i] (IN|TO) [m_j]".

- If the only dependency labels in the dependency path between the head words of m_i and m_j are "prep" (prepositional modifier), accept the mention pair.

3.4 Formulaic Structures

- The ACE-2004 annotator guidelines specify that several relations such as reporter signing off, addresses, etc. are often specified in standard structures. We check for the existence of patterns such as "[m_i] / [m_j]", "[m_i] [m_j]",

Category	Feature
For every word w_k in mention m_i	POS of w_k and offset from lw
	w_k and offset from lw
	POS of w_k, w_k, and offset from lw
	POS of w_k, offset from lw, and lw
	$Bc(w_k)$ and offset from lw
	POS of w_k, $Bc(w_k)$, and offset from lw
	POS of w_k, offset from lw, and $Bc(lw)$
Contextual	$C_{-1,-1}$ of m_i
	$C_{+1,+1}$ of m_i
	$P_{-1,-1}$ of m_i
	$P_{+1,+1}$ of m_i
NE tags	tag of NE, if lw of NE coincides with lw of m_i in the sentence
Syntactic parse	parse-label of parse tree constituent that exactly covers m_i
	parse-labels of parse tree constituents covering m_i

Table 2: Features used in our mention entity typing (MET) system. The abbreviations are as follows. lw: last word in the mention; $Bc(w)$: the brown cluster bit string representing w; NE: named entity

and whether they satisfy certain semantic entity type constraints.

4 Mention Extraction System

As part of our experiments, we perform RE using predicted mentions. We first describe the features (an overview is given in Table 2) and then describe how we extract candidate mentions from sentences during evaluation.

4.1 Mention Extraction Features

Features for every word in the mention For every word w_k in a mention m_i, we extract seven features. These are a combination of w_k itself, its POS tag, and its integer offset from the last word (lw) in the mention. For instance, given the mention "the operation room", the offsets for the three words in the mention are -2, -1, and 0 respectively. These features are meant to capture the word and POS tag sequences in mentions.

We also use word clusters which are automatically generated from unlabeled texts, using the Brown clustering (Bc) algorithm of (Brown et al., 1992). This algorithm outputs a binary tree where words are leaves in the tree. Each word (leaf) in the tree can be represented by its unique path from the root and this path can be represented as a simple bit string. As part of our features, we use the cluster bit string representation of w_k and lw.

Category	Feature
POS	POS of single word between m_1, m_2
	hw of m_i, m_j and $P_{-1,-1}$ of m_i, m_j
	hw of m_i, m_j and $P_{-1,-1}$ of m_i, m_j
	hw of m_i, m_j and $P_{+1,+1}$ of m_i, m_j
	hw of m_i, m_j and $P_{-2,-1}$ of m_i, m_j
	hw of m_i, m_j and $P_{-1,+1}$ of m_i, m_j
	hw of m_i, m_j and $P_{+1,+2}$ of m_i, m_j
Base chunk	any base phrase chunk between m_i, m_j

Table 3: Additional RE features.

Contextual We extract the word $C_{-1,-1}$ immediately before m_i, the word $C_{+1,+1}$ immediately after m_i, and their associated POS tags P.

NE tags We automatically annotate the sentences with named entity (NE) tags using the named entity tagger of (Ratinov and Roth, 2009). This tagger annotates *proper nouns* with the tags PER (person), ORG (organization), LOC (location), or MISC (miscellaneous). If the lw of m_i coincides (actual token offset) with the lw of any NE annotated by the NE tagger, we extract the NE tag as a feature.

Syntactic parse We parse the sentences using the syntactic parser of (Klein and Manning, 2003). We extract the label of the parse tree constituent (if it exists) that *exactly covers* the mention, and also labels of all constituents that *covers* the mention.

4.2 Extracting Candidate Mentions

From a sentence, we gather the following as candidate mentions: all nouns and possessive pronouns, all named entities annotated by the the NE tagger (Ratinov and Roth, 2009), all base noun phrase (NP) chunks, all chunks satisfying the pattern: NP (PP NP)+, all NP constituents in the syntactic parse tree, and from each of these constituents, all substrings consisting of two or more words, provided the substrings do not start nor end on punctuation marks. These mention candidates are then fed to our mention entity typing (MET) classifier for type prediction (more details in Section 6.3).

5 Relation Extraction System

We build a supervised RE system using sentences annotated with entity mentions and predefined target relations. During evaluation, when given a pair of mentions m_i, m_j, the system predicts whether any of the predefined target relation holds between the mention pair.

Most of our features are based on the work of (Zhou et al., 2005; Chan and Roth, 2010). Due to space limitations, we refer the reader to our prior work (Chan and Roth, 2010) for the lexical, structural, mention-level, entity type, and dependency features. Here, we only describe the features that were not used in that work.

As part of our RE system, we need to extract the head word (hw) of a mention (m), which we heuristically determine as follows: if m contains a preposition and a noun preceding the preposition, we use the noun as the hw. If there is no preposition in m, we use the last noun in m as the hw.

POS features If there is a single word between the two mentions, we extract its POS tag. Given the hw of m, $P_{i,j}$ refers to the sequence of POS tags in the immediate context of hw (we exclude the POS tag of hw). The offsets *i* and *j* denote the position (relative to hw) of the first and last POS tag respectively. For instance, $P_{-2,-1}$ denotes the sequence of two POS tags on the immediate left of hw, and $P_{-1,+1}$ denotes the POS tag on the immediate left of hw and the POS tag on the immediate right of hw.

Base phrase chunk We add a boolean feature to detect whether there is any base phrase chunk in the text span between the two mentions.

6 Experiments

We use the ACE-2004 dataset (catalog LDC2005T09 from the Linguistic Data Consortium) to conduct our experiments. Following prior work, we use the news wire (nwire) and broadcast news (bnews) corpora of ACE-2004 for our experiments, which consists of 345 documents.

To build our RE system, we use the LIBLINEAR (Fan et al., 2008) package, with its default settings of L2-loss SVM (dual) as the solver, and we use an epsilon of 0.1. To ensure that this baseline RE system based on the features in Section 5 is competitive, we compare against the state-of-the-art feature-based RE systems of (Jiang and Zhai, 2007) and (Chan and Roth, 2010). In these works, the authors reported performance on undirected coarse-grained RE. Performing 5-fold cross validation on the nwire and bnews corpora, (Jiang and Zhai, 2007) and (Chan and Roth, 2010) reported F-measures of 71.5 and 71.2, respectively. Using the same evaluation setting, our baseline RE system achieves a competitive 71.4 F-measure.

We build three RE classifiers: *binary*, *coarse*, *fine*. Lumping all the predefined target relations into a single label, we build a *binary* classifier to predict whether any of the predefined relations exists between a given mention pair.

In this work, we model the argument order of the mentions when performing RE, since relations are usually asymmetric in nature. For instance, we consider m_i:EMP-ORG:m_j and m_j:EMP-ORG:m_i to be distinct relation types. In our experiments, we extracted a total of 55,520 examples or mention pairs. Out of these, 4,011 are positive relation examples annotated with 6 coarse-grained relation types and 22 fine-grained relation types[5].

We build a *coarse-grained* classifier to disambiguate between 13 relation labels (two asymmetric labels for each of the 6 coarse-grained relation types and a null label). We similarly build a *fine-grained* classifier to disambiguate between 45 relation labels.

6.1 Evaluation Method

For our experiments, we adopt the experimental setting in our prior work (Chan and Roth, 2010) of ensuring that all examples from a single document are either all used for training, or all used for evaluation.

In that work, we also highlight that ACE annotators rarely duplicate a relation link for coreferent mentions. For instance, assume mentions m_i, m_j, and m_k are in the same sentence, mentions m_i and m_j are coreferent, and the annotators tag the mention pair m_j, m_k with a particular relation r. The annotators will rarely duplicate the same (implicit)

[5]We omit a single relation: Discourse (DISC). The ACE-2004 annotation guidelines states that the DISC relation is established only for the purposes of the discourse and does not reference an official entity relevant to world knowledge. In this work, we focus on semantically meaningful relations. Furthermore, the DISC relation is dropped in ACE-2005.

RE model	10 documents			5% of data			80% of data		
	Rec%	Pre%	F1%	Rec%	Pre%	F1%	Rec%	Pre%	F1%
Binary	58.0	80.3	67.4	64.4	80.6	71.6	73.2	84.0	78.2
Binary+Patterns	73.1	78.5	**75.7** (+8.3)	75.3	80.6	**77.9**	80.1	84.2	**82.1**
Coarse	33.5	62.5	43.6	42.4	66.2	51.7	62.1	75.5	68.1
Coarse+Patterns	44.2	59.6	**50.8** (+7.2)	51.2	64.2	**56.9**	68.0	75.4	**71.5**
Fine	18.1	47.0	26.1	26.3	51.6	34.9	51.6	68.4	58.8
Fine+Patterns	24.8	43.5	**31.6** (+5.5)	32.2	48.9	**38.9**	56.4	67.5	**61.5**

Table 4: Micro-averaged (across the 5 folds) RE results using gold mentions.

RE model	10 documents			5% of data			80% of data		
	Rec%	Pre%	F1%	Rec%	Pre%	F1%	Rec%	Pre%	F1%
Binary	32.2	46.6	38.1	35.5	48.9	41.1	40.1	52.7	45.5
Binary+Patterns	46.7	45.9	**46.3** (+8.2)	47.6	47.8	**47.2**	50.2	50.4	**50.3**
Coarse	18.6	41.1	25.6	22.4	40.9	28.9	32.3	47.5	38.5
Coarse+Patterns	26.8	34.7	**30.2** (+4.6)	30.3	37.0	**33.3**	38.9	42.9	**40.8**
Fine	10.7	32.2	16.1	14.6	33.4	20.3	26.9	44.3	33.5
Fine+Patterns	15.7	26.3	**19.7** (+3.6)	19.4	29.2	**23.3**	31.7	38.3	**34.7**

Table 5: Micro-averaged (across the 5 folds) RE results using predicted mentions.

relation r between m_i and m_k, thus leaving the gold relation label as null. Whether this is correct or not is debatable. However, to avoid being penalized when our RE system actually correctly predicts the label of an implicit relation, we take the following approach.

During evaluation, if our system correctly predicts an implicit label, we simply switch its prediction to the null label. Since the RE recall scores only take into account non-null relation labels, this scoring method does not change the recall, but could marginally increase the precision scores by decreasing the count of RE predictions. In our experiments, we observe that both the usual and our scoring method give very similar RE results and the experimental trends remain the same. Of course, using this scoring method requires coreference information, which is available in the ACE data.

6.2 RE Evaluation Using Gold Mentions

To perform our experiments, we split the 345 documents into 5 equal sets. In each of the 5 folds, 4 sets (276 documents) are reserved for drawing training examples, while the remaining set (69 documents) is used as evaluation data. In the experiments described in this section, we use the gold mentions available in the data.

When one only has a small amount of training data, it is crucial to take advantage of external knowledge such as the syntactico-semantic structures. To simulate this setting, in each fold, we randomly selected 10 documents from the fold's available training documents (about 3% of the total 345 documents) as training data. We built one binary, one coarse-grained, and one fine-grained classifier for each fold.

In Section 2, we described how we trained a baseline RE classifier (RE_{base}) and a RE classifier using the syntactico-semantic patterns (RE_s).

We first apply RE_{base} on each test example mention pair (m_i,m_j) to obtain the RE baseline results, showing these in Table 4 under the column "10 documents", and in the rows "Binary", "Coarse", and "Fine". We then applied RE_s on the test examples as described in Section 2, showing the results in the rows "Binary+Patterns", "Coarse+Patterns", and "Fine+Patterns". The results show that by using syntactico-semantic structures, we obtain significant F-measure improvements of **8.3**, **7.2**, and **5.5** for binary, coarse-grained, and fine-grained relation predictions respectively.

6.3 RE Evaluation Using Predicted Mentions

Next, we perform our experiments using predicted mentions. ACE-2004 defines 7 coarse-grained *entity* types, each of which are then refined into 43 fine-

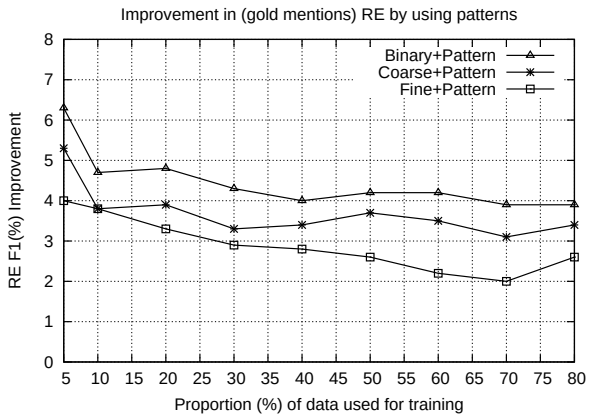

Figure 3: Improvement in (gold mention) RE.

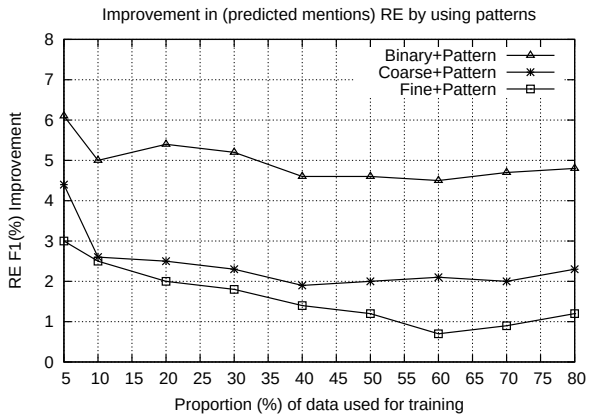

Figure 4: Improvement in (predicted mention) RE.

grained *entity* types. Using the ACE data annotated with mentions and predefined entity types, we build a fine-grained mention entity typing (MET) classifier to disambiguate between 44 labels (43 fine-grained and a null label to indicate not a mention). To obtain the coarse-grained entity type predictions from the classifier, we simply check which coarse-grained type the fine-grained prediction belongs to. We use the LIBLINEAR package with the same settings as earlier specified for the RE system. In each fold, we build a MET classifier using all the (276) training documents in that fold.

We apply RE_{base} on all mention pairs (m_i,m_j) where both m_i and m_j have non null entity type predictions. We show these baseline results in the Rows "Binary", "Coarse", and "Fine" of Table 5.

In Section 2, we described our algorithmic approach (Figure 2) that takes advantage of the structures with predicted mentions. We show the results of this approach in the Rows "Binary+Patterns", "Coarse+Patterns", and "Fine+Patterns" of Table 5. The results show that by leveraging syntactico-semantic structures, we obtain significant F-measure improvements of **8.2**, **4.6**, and **3.6** for binary, coarse-grained, and fine-grained relation predictions respectively.

7 Analysis

We first show statistics regarding the syntactico-semantic structures. In Section 3, we mentioned that ACE-2004 identified five types of structures: premodifier, possessive, preposition, formulaic, and verbal. On the 4,011 examples that we experimented on, premodifiers are the most frequent, accounting for 30.5% of the examples (or about 1,224 examples). The occurrence distributions of the other structures are 18.9% (possessive), 23.9% (preposition), 7.2% (formulaic), and 19.5% (verbal). Hence, the four syntactico-semantic structures that we focused on in this paper account for a large majority (80%) of the relations.

Pattern type	Rec%	Pre%
PreMod	86.8	79.7
Poss	94.3	88.3
Prep	94.6	20.0
Formula	85.5	62.2

Table 6: Recall and precision of the patterns.

In Section 6, we note that out of 55,520 mention pairs, only 4,011 exhibit valid relations. Thus, the proportion of positive relation examples is very sparse at 7.2%. If we can effectively identify and discard most of the negative relation examples, it should improve RE performance, including yielding training data with a more balanced label distribution.

We now analyze the utility of the patterns. As shown in Table 6, the patterns are effective in inferring the structure of mention pairs. For instance, applying the premodifier patterns on the 55,520 mention pairs, we correctly identified 86.8% of the 1,224 premodifier occurrences as premodifiers, while incurring a false-positive rate of only about 20%[6]. We

[6]Random selection will give a precision of about 2.2% (1,224 out of 55,520) and thus a false-positive rate of 97.8%

note that preposition structures are relatively harder to identify. Some of the reasons are due to possibly multiple prepositions in between a mention pair, preposition sense ambiguity, pp-attachment ambiguity, etc. However, in general, we observe that inferring the structures allows us to discard a large portion of the mention pairs which have no valid relation between them. The intuition behind this is the following: if we infer that there is a syntactico-semantic structure between a mention pair, then it is likely that the mention pair exhibits a valid relation. Conversely, if there is a valid relation between a mention pair, then it is likely that there exists a syntactico-semantic structure between the mentions.

Next, we repeat the experiments in Section 6.2 and Section 6.3, while gradually increasing the amount of training data used for training the RE classifiers. The detailed results of using 5% and 80% of all available data are shown in Table 4 and Table 5. Note that these settings are with respect to all 345 documents and thus the 80% setting represents using all 276 training documents in each fold. We plot the intermediate results in Figure 3 and Figure 4. We note that leveraging the structures provides improvements on all experimental settings. Also, intuitively, the *binary* predictions benefit the most from leveraging the structures. How to further exploit this is a possible future work.

8 Conclusion

In this paper, we propose a novel algorithmic approach to RE by exploiting syntactico-semantic structures. We show that this approach provides several advantages and improves RE performance. There are several interesting directions for future work. There are probably many near misses when we apply our structure patterns on predicted mentions. For instance, for both premodifier and possessive structures, we require that one mention completely includes the other. Relaxing this might potentially recover additional valid mention pairs and improve performance. We could also try to learn classifiers to automatically identify and disambiguate between the different syntactico-semantic structures. It will also be interesting to feedback the predictions of the structure patterns to the mention entity typing classifier and possibly retrain to obtain a better classifier.

Acknowledgements This research is supported by the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181. Any opinions, findings, and conclusion or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the view of the DARPA, AFRL, or the US government.

We thank Ming-Wei Chang and Quang Do for building the mention extraction system.

References

Peter F. Brown, 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.

Yee Seng Chan and Dan Roth. 2010. Exploiting background knowledge for relation extraction. In *Proceedings of the International Conference on Computational Linguistics (COLING)*, pages 152–160.

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

Katrin Fundel, Robert Küffner, and Ralf Zimmer. 2007. Relex – Relation extraction using dependency parse trees. *Bioinformatics*, 23(3):365–371.

Mark A. Greenwood and Mark Stevenson. 2006. Improving semi-supervised acquisition of relation extraction patterns. In *Proceedings of the COLING-ACL Workshop on Information Extraction Beyond The Document*, pages 29–35.

Jing Jiang and ChengXiang Zhai. 2007. A systematic exploration of the feature space for relation extraction. In *Proceedings of Human Language Technologies - North American Chapter of the Association for Computational Linguistics (HLT-NAACL)*, pages 113–120.

Jing Jiang. 2009. Multi-task transfer learning for weakly-supervised relation extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 1012–1020.

Nanda Kambhatla. 2004. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 178–181.

Dan Klein and Christoper D. Manning. 2003. Fast exact inference with a factored model for natural language parsing. In *The Conference on Advances in Neural Information Processing Systems (NIPS)*, pages 3–10.

Longhua Qian, Guodong Zhou, Qiaomin Zhu, and Peide Qian. 2008. Relation extraction using convolution tree kernel expanded with entity features. In *Pacific Asia Conference on Language, Information and Computation*, pages 415–421.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Annual Conference on Computational Natural Language Learning (CoNLL)*, pages 147–155.

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

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

Jun Zhu, Zaiqing Nie, Xiaojiang Liu, Bo Zhang, and Ji-Rong Wen. 2009. Statsnowball: a statistical approach to extracting entity relationships. In *The International World Wide Web Conference*, pages 101–110.