Modeling Semantic Relations Expressed by Prepositions

Vivek Srikumar and **Dan Roth**
University of Illinois, Urbana-Champaign
Urbana, IL. 61801.
{vsrikum2, danr}@illinois.edu

Abstract

This paper introduces the problem of predicting semantic relations expressed by prepositions and develops statistical learning models for predicting the relations, their arguments and the semantic types of the arguments. We define an inventory of 32 relations, building on the word sense disambiguation task for prepositions and collapsing related senses across prepositions. Given a preposition in a sentence, our computational task to jointly model the preposition relation and its arguments along with their semantic types, as a way to support the relation prediction. The annotated data, however, only provides labels for the relation label, and not the arguments and types. We address this by presenting two models for preposition relation labeling. Our generalization of latent structure SVM gives close to 90% accuracy on relation labeling. Further, by jointly predicting the relation, arguments, and their types along with preposition sense, we show that we can not only improve the relation accuracy, but also significantly improve sense prediction accuracy.

1 Introduction

This paper addresses the problem of predicting semantic relations conveyed by prepositions in text. Prepositions express many semantic relations between their governor and object. Predicting these can help advancing text understanding tasks like question answering and textual entailment. Consider the sentence:

(1) The book of Prof. Alexander on primary school methods is a valuable teaching resource.

Here, the preposition *on* indicates that *the book* and *primary school methods* are connected by the relation `Topic` and *of* indicates the `Creator-Creation` relation between *Prof. Alexander* and *the book*. Predicting these relations can help answer questions about the subject of the book and also recognize the entailment of sentences like *Prof. Alexander has written about primary school methods.*

Being highly polysemous, the same preposition can indicate different kinds of relations, depending on its governor and object. Furthermore, several prepositions can indicate the same semantic relation. For example, consider the sentence:

(2) Poor care led to her death from pneumonia.

The preposition *from* in this sentence expresses the relation `Cause`(*death*, *pneumonia*). In a different context, it can denote other relations, as in the phrases *copied from the film* (`Source`) and *recognized from the start* (`Temporal`). On the other hand, the relation `Cause` can be expressed by several prepositions; for example, the following phrases express a `Cause` relation: *died of pneumonia* and *tired after the surgery*.

We characterize semantic relations expressed by transitive prepositions and develop accurate models for predicting the relations, identifying their arguments and recognizing the semantic types of the arguments. Building on the word sense disambiguation task for prepositions, we collapse semantically related senses across prepositions to derive our relation inventory. These relations act as predicates in a predicate-argument representation, where the arguments are the governor and the object of the

Transactions of the Association for Computational Linguistics, 1 (2013) 231–242. Action Editor: Noah Smith.
Submitted 11/2012; Revised 2/2013; Published 5/2013.

preposition. While ascertaining the arguments is a largely syntactic decision, we point out that syntactic parsers do not always make this prediction correctly. However, as illustrated in the examples above, identifying the relation depends on the governor and object of the preposition.

Given a sentence and a preposition, our goal is to model the predicate (i.e. the preposition relation) and its arguments (i.e. the governor and object). Very often, the relation label is not influenced by the surface form of the arguments but rather by their *semantic types*. In sentence (2) above, we want the predicate to be `Cause` when the object of the preposition is any illness. We thus suggest to model the argument types along with the preposition relations and arguments, using different notions of types. These three related aspects of the relation prediction task are further explained in Section 3 leading up to the problem definition.

Though we wish to predict relations, arguments and types, there is no corpus which annotates all three. The SemEval 2007 shared task of word sense disambiguation for prepositions provides sense annotations for prepositions. We use this data to generate training and test corpora for the relation labels. In Section 4, we present two models for the prepositional relation identification problem. The first model considers all possible argument candidates from various sources along with all argument types to predict the preposition relation label. The second model treats the arguments and types as latent variables during learning using a generalization of the latent structural SVM of (Yu and Joachims, 2009). We show in Section 5 that this model not only predicts the arguments and types, but also improves relation prediction performance.

The primary contributions of this paper are:

1. We introduce a new inventory of preposition relations that covers the 34 prepositions that formed the basis of the SemEval 2007 task of preposition sense disambiguation.

2. We model preposition relations, arguments and their types jointly and propose a learning algorithm that learns to predict all three using training data that annotates only relation labels.

3. We show that jointly predicting relations with word sense not only improves the relation predictor, but also gives a significant improvement in sense prediction.

2 Prepositions & Predicate-Argument Semantics

Semantic role labeling (cf. (Gildea and Jurafsky, 2002; Palmer et al., 2010; Punyakanok et al., 2008) and others) is the task of converting text into a predicate-argument representation. Given a trigger word or phrase in a sentence, this task solves two related prediction problems: (a) identifying the relation label, and (b) identifying and labeling the arguments of the relation.

This problem has been studied in the context of verb and nominal triggers using the PropBank (Palmer et al., 2005) and NomBank (Meyers et al., 2004) annotations over the Penn Treebank, and also using the FrameNet lexicon (Fillmore et al., 2003), which allows arbitrary words to trigger semantic frames.

This paper focuses on semantic relations expressed by transitive prepositions[1]. We can define the two prediction tasks for prepositions as follows: identifying the relation label for a preposition, and predicting the arguments of the relation. Prepositions can mark arguments (both core and adjunct) for verbal and nominal predicates. In addition, they can also trigger relations that are not part of other predicates. For example, in sentence (3) below, the prepositional phrase starting with *to* is an argument of the verb *visit*, but the *in* triggers an independent relation indicating the location of the aquarium.

(3) The children enjoyed the visit to the aquarium in Coney Island.

FrameNet covers some prepositional relations, but allows only temporal, locative and directional senses of prepositions to evoke frames, accounting for only 3% of the targets in the SemEval 2007 shared task of FrameNet parsing. In fact, the state-of-the-art FrameNet parser of (Das et al., 2010) does not consider any frame inducing prepositions.

(Baldwin et al., 2009) highlights the importance of studying prepositions for a complete linguistic

[1] By transitive prepositions we refer to the standard usage of prepositions that take an object. In particular, we do not consider prepositional particles in our analysis.

analysis of sentences and surveys work in the NLP literature that addresses the syntax and semantics of prepositions. One line of work (Ye and Baldwin, 2006) addressed the problem of preposition semantic role labeling by considering prepositional phrases that act as arguments of verbs according to the PropBank annotation. They built a system that predicts the labels of these prepositional phrases alone. However, by definition, this covered only verb-attached prepositions. (Zapirain et al., 2012) studied the impact of automatically learned selectional preferences for predicting arguments of verbs and showed that modeling prepositional phrases separately improves the performance of argument prediction.

Preposition semantics has also been studied via the Preposition Project (Litkowski and Hargraves, 2005) and the related SemEval 2007 shared task of word sense disambiguation of prepositions (Litkowski and Hargraves, 2007). The Preposition Project identifies preposition senses based on their definitions in the Oxford Dictionary of English. There are 332 different labels to be predicted with a wide variance in the number of senses per preposition ranging from 2 (*during* and *as*) to 25 (*on*). For example, according to the preposition sense inventory, the preposition *from* in sentence (2) above will be labeled with the sense **from:12(9)** to indicate a cause. (Dahlmeier et al., 2009) added sense annotation to seven prepositions in four sections of the Penn Treebank with the goal of studying their interaction with verb arguments.

Using the SemEval data, (Tratz and Hovy, 2009) and (Hovy et al., 2010) showed that the arguments offer an important cue to identify the sense of the preposition and (Tratz, 2011) showed further improvements by refining the sense inventory. However, though these works used a dependency parser to identify arguments, in order to overcome parsing errors, they augment the parser's predictions using part-of-speech based heuristics.

We argue that, while disambiguating the sense of a preposition does indeed reveal nuances of its meaning, it leads to a proliferation of labels to be predicted. Most importantly, sense labels do not transfer to other prepositions that express the same meaning. For example, both *finish lunch before noon* and *finish lunch by noon* express a `Temporal` relation. According to the Preposition Project, the sense label for the first preposition is **before:1(1)**, and that for the second is **by:17(4)**. This both defeats the purpose of identifying the relations to aid natural language understanding and makes the prediction task harder than it should be: using the standard word sense classification approach, we need to train a separate classifier for each word because the labels are defined per-preposition. In other words, we cannot share features across the different prepositions. This motivates the need to combine such senses of prepositions into the same class label.

In this direction, (O'Hara and Wiebe, 2009) describes an inventory of preposition relations obtained using Penn Treebank function tags and frame elements from FrameNet. (Srikumar and Roth, 2011) merged preposition senses of seven prepositions into relation labels. (Litkowski, 2012) also suggests collapsing the definitions of prepositions into a smaller set of semantic classes. To aid better generalization and to reduce the label complexity, we follow this line of work to define a set of relation labels which abstract word senses across prepositions[2].

3 Preposition-triggered Relations

This section describes the inventory of preposition relations introduced in this paper, and then identifies the components of the preposition relation extraction problem.

3.1 Preposition Relation Inventory

We build our relation inventory using the sense annotation in the Preposition Project, focusing on the 34 prepositions[3] annotated for the SemEval-2007 shared task of preposition sense disambiguation.

As discussed in Section 2, we construct the inventory of preposition relations by collapsing semantically related preposition senses across differ-

[2]Since the preposition sense data is annotated over FrameNet sentences, sense annotation can be used to extend FrameNet (Litkowski, 2012). We believe that the abstract labels proposed in this paper can further help in this effort.

[3]We consider the following prepositions: about, above, across, after, against, along, among, around, as, at, before, behind, beneath, beside, between, by, down, during, for, from, in, inside, into, like, of, off, on, onto, over, round, through, to, towards, and with. This does not include multi-word prepositions such as *because of* and *due to*.

ent prepositions. For each sense that is defined, the Preposition Project also specifies related prepositions. These definitions and related prepositions provide a starting point to identify senses that can be merged across prepositions. We followed this with a manual cleanup phase. Some senses do not cleanly align with a single relation because the definitions include idiomatic or figurative usage. For example, the sense **in:7(5)** of the preposition *in*, according to the definition, includes both spatial and figurative notions of the spatial sense (that is, both *in London* and *in a film*). In such cases, we sampled 20 examples from the SemEval 2007 training set and assigned the relation label based on majority. If sufficient examples could not be sampled, these senses were added to the label `Other`, which is not a semantically coherent category and represents the 'overflow' case.

Overall, we have 32 labels, which are listed in Table 1[4]. A companion publication (available on the authors' website) provides detailed definitions of each relation and the senses that were merged to create each label. Since we define relations to be groups of preposition sense labels, each sense can be uniquely mapped to a relation label. Hence, we can use the annotated sense data from SemEval 2007 to obtain a corpus of relation-labeled sentences.

To validate the labeling scheme, two native speakers of English annotated 200 sentences from the SemEval training corpus using only the definitions of the labels as the annotation guidelines. We measured Cohen's kappa coefficient (Cohen, 1960) between the annotators to be 0.75 and also between each annotator and the original corpus to be 0.76 and 0.74 respectively.

3.2 Preposition Relation Extraction

The input to the prediction problem consists of a preposition in a sentence and the goal is to jointly model the following: (i) The relation expressed by the preposition, and (ii) The arguments of the relation, namely the governor and the object.

We use sentence (2) in the introduction as our running example the following discussion. In our running example, the relation label is `Cause`. We represent the predicted relation label by r.

Relation Name	**Example**
Activity	good at boxing
Agent	opened by Annie
Attribute	walls of stone
Beneficiary	fight for Napoleon
Cause	died of cancer
Co-Participants	pick one among these
Destination	leaving for London
Direction	drove towards the border
EndState	driven to tears
Experiencer	warm towards her
Instrument	cut with a knife
Journey	travel by road
Location	living in London
Manner	scream like an animal
MediumOfCommunication	new show on TV
Numeric	increase by 10%
ObjectOfVerb	murder of the boys
Opponent/Contrast	fight with him
Other	*all others*
Participant/Accompanier	steak with wine
PartWhole	member of gang
PhysicalSupport	lean against the wall
Possessor	son of a friend
ProfessionalAspect	works in publishing
Purpose	tools for making it
Recipient	unkind to her
Separation	ousted from power
Source	purchased from the shop
Species	city of Prague
StartState	recover from illness
Temporal	arrived on Monday
Topic	books on Shakespeare

Table 1: List of preposition relations

Arguments The relation label crucially depends on correctly identifying the arguments of the preposition, which are *death* and *pneumonia* in our running example. While a parser can identify the arguments of a preposition, simply relying on the parser may impose an upper limit on the accuracy of relation prediction.

We build an oracle experiment to highlight this limitation. Table 2 shows the recall of the easy-first dependency parser of (Goldberg and Elhadad, 2010) on Section 23 of the Penn Treebank for identifying the governor and object of prepositions.

We define heuristics that generate a candidate governors and objects for a preposition. For the gov-

[4]Note that, even though we do not consider intransitive prepositions, the definitions of some relations in Table 1 could be extended apply to prepositional particles such *drive down* (`Direction`) and *run about* (`Manner`).

ernor, this set includes the previous verb or noun and for the object, it includes only the next noun. The row labeled Best(Parser, Heuristics) shows the performance of an oracle predictor which selects the true governor/object if present among the parser's prediction and the heuristics. We see that, even for the in-domain case, if we are able to re-rank the candidates, we could achieve a big improvement in argument identification.

	Recall	
	Governor	**Object**
Parser	88.88	92.37
Best(Parser, Heuristics)	92.50	93.06

Table 2: Identifying governor and object of prepositions in the Penn Treebank data. Here, Best(Parser, Heuristics) reports the performance of an oracle that picks the true governor and object, if present among the candidates presented by the parser and the heuristic. This presents an in-domain upper bound for governor and object detection. See text for further details.

To overcome erroneous parser decisions, we entertain governor and object candidates proposed both by the parser and the heuristics. In the following discussion, we denote the chosen governor and object by g and o respectively.

Argument types While the primary purpose of this work is to model preposition relations and their arguments, the relation prediction is strongly dependent on the *semantic type* of the arguments. To illustrate this, consider the following incomplete sentence: *The message was delivered at* $\cdots$. This preposition can express both a `Temporal` or a `Location` relation depending on the object (for example, *noon* vs. *the doorstep*).

(Agirre et al., 2008) shows that modeling the semantic type of the arguments jointly with attachment can improve PP attachment accuracy. In this work, we point out that argument types should be modeled jointly with both aspects of the problem of preposition relation labeling.

Types are an abstraction that capture common properties of groups of entities. For example, WordNet provides generalizations of words in the form of their hypernyms. In our running example, we wish to generalize the relation label for *death from pneumonia* to include cases such as *suffering from flu*. Figure 1 shows the hypernym hierarchy for the word *pneumonia*. In this case, synsets in the hypernym hierarchy, like *pathological state* or *physical condition*, would also include ailments like flu.

```
pneumonia
  => respiratory disease
    => disease
      => illness
        => ill health
          => pathological state
            => physical condition
              => condition
                => state
                  => attribute
                    => abstraction
                      => entity
```

Figure 1: Hypernym hierarchy for the word *pneumonia*

We define a semantic type to be a cluster of words. In addition to WordNet hypernyms, we also cluster verbs, nouns and adjectives using the dependency-based word similarity of (Lin, 1998) and treat cluster membership as types. These are described in detail in Section 5.1.

Relation prediction involves not only identifying the arguments, but also selecting the right semantic type for them, which together, help predicting the relation label. Given an argument candidate and a collection of possible types (given by WordNet or the similarity based clusters), we need to select one of the types. For example, in the WordNet case, we need to pick one of the hypernyms in the hypernym hierarchy. Thus, for the governor and object, we have a set of type labels, comprised of one element for each type category. We denote this by $\mathbf{t}^g$ (governor type) and $\mathbf{t}^o$ (object type) respectively.

3.3 Problem definition

The input to our prediction task is a preposition in a sentence. Our goal is to jointly model the relation it expresses, the governor and the object of the relation and the types of each argument (both WordNet hypernyms and cluster membership). We denote the input by $\mathbf{x}$, which consists not only of the preposition but also a set of candidates for the governor and the object and, for each type category, the list of types for the governor and candidate.

The prediction, which we denote by $\mathbf{y}$, consists of the relation r, which can be one of the valid relation labels in Table 1 and the governor and object, denoted by g and o, each of which is one of text segments proposed by the parser or the heuristics. Additionally, $\mathbf{y}$ also consists of type predictions for the governor and object, denoted by $\mathbf{t}^g$ and $\mathbf{t}^o$ respectively, each of which is a vector of labels, one for each type category. Table 3 summarizes the notation described above. We refer to the i^{th} element of vectors using subscripts and use the superscript * to denote gold labels. Recall that we have gold labels only for the relation labels and not for arguments and their types.

Symbol	Meaning
$\mathbf{x}$	Input (pre-processed sentence and preposition)
r	relation label for the preposition
g, o	governor and object of the relation
$\mathbf{t}^g, \mathbf{t}^o$	vectors of type assignments for governor and object respectively
$\mathbf{y}$	Full structure $(r, g, o, \mathbf{t}^g, \mathbf{t}^o)$

Table 3: Summary of notation

4 Learning preposition relations

A key challenge in modeling preposition relations is that our training data only annotates the relation labels and not the arguments and types. In this section, we introduce two approaches for predicting preposition relations using this data.

4.1 Feature Representation

We use the notation $\Phi(\mathbf{x}, \mathbf{y})$ to indicate the feature function for an input $\mathbf{x}$ and the full output $\mathbf{y}$. We build Φ using the features of the components of $\mathbf{y}$:

1. **Arguments**: For g and o, which represent an assignment to the governor and object, we denote the features extracted from the arguments as $\phi_A(\mathbf{x}, g)$ and $\phi_A(\mathbf{x}, o)$ respectively.

2. **Types**: Given a type assignment t_i^g to the i^{th} type category of the governor, we define features $\phi_T(\mathbf{x}, g, t_i^g)$. Similarly, we define features $\phi_T(\mathbf{x}, o, t_i^o)$ for the types of the object.

We combine the argument and their type features to define the features for classifying the relation, which we denote by $\phi(\mathbf{x}, g, o, \mathbf{t}^g, \mathbf{t}^o)$:

$$\phi = \sum_{a \in \{g,o\}} \left(\phi_A(\mathbf{x}, a) + \sum_i \phi_T(\mathbf{x}, a, t_i^a) \right) \quad (1)$$

Section 5 describes the actual features used in our experiments.

Observe that given the arguments and their types, the task of predicting relations is simply a multiclass classification problem. Thus, following the standard convention for multiclass classification, the overall feature representation for the relation *and* argument prediction is defined by conjoining the relation r with features for the corresponding arguments and types, ϕ. This gives us the full feature representation, $\Phi(\mathbf{x}, \mathbf{y})$.

4.2 Model 1: Predicting only relations

The first model aims at predicting only the relation labels and not the arguments and types. This falls into the standard multiclass classification setting, where we wish to predict one of 32 labels. To do so, we sum over all the possible assignments to the rest of the structure and define features for the inputs as

$$\hat{\phi}(\mathbf{x}) = \sum_{g,o,\mathbf{t}^g,\mathbf{t}^o} \phi(\mathbf{x}, g, o, \mathbf{t}^g, \mathbf{t}^o) \quad (2)$$

Effectively, doing so uses all the governor and object candidates and all their semantic types to get a feature representation for the relation classification problem. Once again, for a relation label r, the overall feature representation is defined by conjoining the relation r with the features for that relation $\hat{\phi}$, which we write as $\phi_R(\mathbf{x}, r)$. Note that this summation is computationally inexpensive in our case because the sum decomposes according to equation (1). With a learned weight vector $\mathbf{w}$, the relation label is predicted as

$$r = \arg\max_{r'} \mathbf{w}^T \phi_R(\mathbf{x}, r') \quad (3)$$

We use a structural SVM (Tsochantaridis et al., 2004) to train a weight vector $\mathbf{w}$ that predicts the relation label as above. The training is parameterized by C, which represents the tradeoff between generalization and the hinge loss.

4.3 Model 2: Learning from partial annotations

In the second model, even though our annotation does not provide gold labels for arguments and types, our goal is to predict them. At inference time, if we had a weight vector $\mathbf{w}$, we could predict the full structure using inference as follows:

$$\mathbf{y} = \arg\max_{\mathbf{y}'} \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}) \quad (4)$$

We propose an iterative learning algorithm to learn this weight vector.

In the following discussion, for a labeled example $(\mathbf{x}, \mathbf{y}^*)$, we refer to the missing part of its structure as $h(\mathbf{y}^*)$. That is, $h(\mathbf{y}^*)$ is the assignment to the arguments of the relation and their types. We use the notation $r(\mathbf{y})$ to denote the relation label specified by a structure $\mathbf{y}$.

Our learning algorithm is closely related to recently developed latent variable based frameworks (Yu and Joachims, 2009; Chang et al., 2010a; Chang et al., 2010b), where the supervision provides only partial annotation. We begin by defining two additional inference procedures:

1. **Latent Inference**: Given a weight vector $\mathbf{w}$ and a partially labeled example $(\mathbf{x}, \mathbf{y}^*)$, we can 'complete' the rest of the structure by inferring the highest scoring assignment to the missing parts. In the algorithm, we call this procedure $LatentInf(\mathbf{w}, \mathbf{x}, \mathbf{y}^*)$, which solves the following maximization problem:

$$\hat{\mathbf{y}} = \begin{array}{ll} \arg\max_{\mathbf{y}} & \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}), \\ \text{s.t.} & r(\mathbf{y}) = r(\mathbf{y}^*). \end{array} \quad (5)$$

2. **Loss augmented inference**: This is a variant of the the standard loss augmented inference for structural SVMs, which solves the following maximization problem for a given $\mathbf{x}$ and *fully labeled* $\mathbf{y}^*$:

$$\arg\max_{\mathbf{y}} \mathbf{w}^T \Phi(\mathbf{x}, \mathbf{y}) + \Delta(\mathbf{y}, \mathbf{y}^*) \quad (6)$$

Here, $\Delta(\mathbf{y}, \mathbf{y}^*)$ denotes the loss function. In the standard structural SVMs, the loss is over the entire structure. In the Latent Structural SVM formulation of (Yu and Joachims, 2009), the loss is defined only over the part of the structure with the gold label. In this work, we use the standard Hamming loss over the entire structure, but scale the loss for the elements of $h(\mathbf{y})$ by a parameter $\alpha < 1$. This is a generalization of the latent structural SVM, which corresponds to the setting $\alpha = 0$. The intuition behind having a non-zero α is that in addition to penalizing the learning algorithm if it violates the annotated part of the structure, we also incorporate a small penalty for the rest of the structure.

Using these two inference procedures, we define the learning algorithm as Algorithm 1. The weight vector is initialized using Model 1. The algorithm then finds the best arguments and types for all examples in the training set (steps 3-5). Doing so gives an estimate of the arguments and types for each example, giving us 'fully labeled' structured data. The algorithm then proceeds to use this data to train a new weight vector using the standard structural SVM with the loss augmented inference listed above (step 6). These two steps are repeated several times. Note that as with the summation in Model 1, solving the inference problems described above is computationally inexpensive.

Algorithm 1 Algorithm for learning Model 2

Input: Examples $D = \{\mathbf{x}_i, r(\mathbf{y}_i^*)\}$, where examples are labeled only with the relation labels.

1: Initialize weight vector $\mathbf{w}$ using Model 1
2: **for** $t = 1, 2, \cdots$ **do**
3: **for** $(\mathbf{x}_i, \mathbf{y}_i^*) \in D$ **do**
4: $\hat{\mathbf{y}}_i \leftarrow LatentInf(\mathbf{w}, \mathbf{x}_i, \mathbf{y}_i^*)$ (Eq. 5)
5: **end for**
6: $\mathbf{w} \leftarrow LearnSSVM(\{\mathbf{x}_i, \hat{\mathbf{y}}_i\})$ with the loss augmented inference of Eq. 6
7: **end for**
8: **return** $\mathbf{w}$

Algorithm 1 is parameterized by C and α. The parameter α controls the extent to which the hypothesized labels according to the previous iteration's weight vector influence the learning.

4.4 Joint inference between preposition senses and relations

By defining preposition relations as disjoint sets of preposition senses, we effectively have a hierarchical relationship between senses and relations. This suggests that joint inference can be employed between sense and relation predictions with a validity constraint connecting the two. The idea of employing inference to combine independently trained predictors to obtain a coherent output structure has been used for various NLP tasks in recent years, starting with the work of (Roth and Yih, 2004; Roth and Yih, 2007).

We use the features defined by (Hovy et al., 2010), which we write as $\phi_s(\mathbf{x}, s)$ for a given input $\mathbf{x}$ and sense label s, and train a separate preposition sense model on the SemEval data with features $\phi_s(\mathbf{x}, s)$ using the structural SVM algorithm. Thus, we have two weight vectors – the one for predicting preposition relations described earlier, and the preposition sense weight vector. At prediction time, for a given input, we find the highest scoring *joint* assignment to the relation, arguments and types and the sense, subject to the constraint that the sense and the relation agree according to the definition of the relations.

5 Experiments and Results

The primary research goal of our experiments is to evaluate the different models (Model 1, Model 2 and joint relation-sense inference) for predicting preposition relations. In additional analysis experiments, we also show that the definition of preposition relations indeed captures cross-preposition semantics by taking advantage of shared features and also highlight the need for going beyond the syntactic parser.

5.1 Types and Features

Types As described in Section 3, we use WordNet hypernyms as one of the type categories. We use all hypernyms within four levels in the hypernym hierarchy for all senses.

The second type category is defined by word-similarity driven clusters. We briefly describe the clustering process here. The thesaurus of (Lin, 1998) specifies similar lexical items for a given word along with a similarity score from 0 to 1. It treats nouns, verbs and adjectives separately. We use the score to cluster groups of similar words using a greedy set-covering approach. Specifically, we randomly select a word which is not yet in a cluster as the center of a new cluster and add all words whose score is greater than σ to it. We repeat this process till all words are in some cluster. A word can appear in more than one cluster because *all* words similar to the cluster center are added to the cluster. We repeat this process for $\sigma \in \{0.1, 0.125, 0.15, 0.175, 0.2, 0.25\}$. By increasing the value of σ, the clusters become more selective and hence smaller. Table 4 shows example noun clusters created using $\sigma = 0.15$. For a given word, identifiers for clusters to which the word belongs serve as type label candidates for this type category[5].

Features Our argument features, denoted by ϕ_A in Section 4.1, are derived from the preposition sense feature set of (Hovy et al., 2010) and extract the following from the argument: 1. Word, part-of-speech, lemma and capitalization indicator, 2. Conflated part-of-speech (one of Noun, Verb, Adjective, Adverb, and Other), 3. Indicator for existence in WordNet, 4. WordNet synsets for the first and all senses, 5. WordNet lemma, lexicographer file names and part, member and substance holonyms, 6. Roget thesaurus divisions for the word, 7. The first and last two and three letters, and 8. Indicators for known affixes. Our type features (ϕ_T) are simply indicators for the type label, conjoined with the type category.

One advantage of abstracting word senses into relations is that we can share features across different prepositions. The base feature set (for both types and arguments) defined above does not encode information about the preposition to be classified. We do so by conjoining the features with the preposition. In addition, since the relation labels are shared across all prepositions, we include the base features as a shared representation between prepositions.

We consider two variants of our feature sets. We refer to the features described above as the **typed** features. In addition, we define the **typed+gen** features by conjoining argument and type features of **typed** with the name of the generator that proposes the argument. Recall that governor candidates are proposed by the dependency parser, or by the heuristics described earlier. Hence, for

[5]The clusters can be downloaded from the authors' website.

Jimmy Carter; Ronald Reagan; richard nixon; George Bush; Lyndon Johnson; Richard M. Nixon; Gerald Ford
metalwork; porcelain; handicraft; jade; bronzeware; carving; pottery; ceramic; earthenware; jewelry; stoneware; lacquerware
degradation; erosion; pollution; logging; desertification; siltation; urbanization; felling; poaching; soil erosion; depletion; water pollution; deforestation
expert; Wall Street analyst; analyst; economist; telecommunications analyst; strategist; media analyst
fox news channel; NBC News; MSNBC; Fox News; CNBC; CNNfn; C-Span
Tuesdays; Wednesdays; weekday; Mondays; Fridays; Thursdays; sundays; Saturdays

Table 4: Examples of noun clusters generated using the set-covering approach for $\sigma = 0.15$

a governor, the **typed+gen** features would conjoin the corresponding **typed** features with one of *parser*, *previous-verb*, *previous-noun*, *previous-adjective*, or *previous-word*.

5.2 Experimental setup and data

All our experiments are based on the SemEval 2007 data for preposition sense disambiguation (Litkowski and Hargraves, 2007) comprising word sense annotation over 16176 training and 8058 examples of prepositions labeled with their senses. We pre-processed sentences with part-of-speech tags using the Illinois POS tagger and dependency graphs using the parser of (Goldberg and Elhadad, 2010)[6]. For the experiments described below, we used the relation-annotated training set to train the models and evaluate accuracy of prediction on the test set.

We chose the structural SVM parameter C using five-fold cross-validation on a 1000 random examples chosen from the training set. For Model 2, we picked $\alpha = 0.1$ using a validation set consisting of a separate set of 1000 training examples. We ran Algorithm 1 for 20 rounds.

Predicting the most frequent relation for a preposition gives an accuracy of 21.18%. Even though the performance of the most-frequent relation label is poor, it does not represent the problem's difficulty and is not a good baseline. To compare, for preposition senses, using features from the neighboring words, (Ye and Baldwin, 2007) obtained an accuracy of 69.3%, and with features designed for the preposition sense task, (Hovy et al., 2010) get up to 84.8% accuracy for the task. Our re-implementation of the latter system using a different set of pre-processing tools gets an accuracy of 83.53%.

For preposition relations, our baseline system for relation labeling uses the **typed** feature set, but without any type information. This produces an accuracy of 88.01% with Model 1 and 88.64% with Model 2. We report statistical significance of results using our implementation of Dan Bikel's stratified-shuffling based statistical significance tester[7].

5.3 Main results: Relation prediction

Our main result, presented in Table 5, compares the baseline model (without types) against other systems, using both models described in Section 4. First, we see that adding type information (**typed**) improves performance over the baseline. Expanding the feature space (**typed+gen**) gives further improvements. Finally, jointly predicting the relations with preposition senses gives another improvement.

Setting	**Accuracy**	
	Model 1	**Model 2**
No types	88.01	88.64
typed	**88.77**	89.14
typed+gen	**89.90***	**89.43***
Joint **typed+gen** & sense	**89.99***	**90.26***†

Table 5: **Main results**: Accuracy of relation labeling. Results in bold are statistically significant ($p < 0.01$) improvements over the system that is unaware of types. Superscripts $*$ and † indicate significant improvements over **typed** and **typed+gen** respectively at $p < 0.01$. For Model 2, the improvement of **typed** over the model without types is significant at $p < 0.05$.

Our objective is not predicting preposition sense. However, we observe that with Model 2, jointly predicting the sense and relations improves not only the performance of relation identification, but via joint inference between relations and senses also leads to a large improvement in sense prediction accuracy. Table 6 shows the accuracy for sense prediction. We

[6]We used the Curator (Clarke et al., 2012) for all pre-processing.

[7]http://www.cis.upenn.edu/~dbikel/software.html

see that while Model 1 does not lead to a significant improvement in the accuracy, Model 2 gives an absolute improvement of over 1%.

Setting	Sense accuracy
Hovy (re-implementation)	83.53
Joint + Model 1	83.78
Joint + Model 2	**84.78***

Table 6: Sense prediction performance. Joint inference with Model 1, while improving relation performance, does not help sense accuracy in comparison to our re-implementation of the Hovy sense disambiguation system. However, with Model 2, the improvement is statistically significant at $p < 0.01$.

5.4 Ablation experiments

Feature sharing across prepositions In our first analysis experiment, we seek to highlight the utility of sharing features between different prepositions. To do so, we compare the performance of a system trained without shared features against the type-independent system, which uses shared features. To discount the influence of other factors, we use Model 1 in the **typed** setting without any types. Table 7 reports the accuracy of relation prediction for these two feature sets. We observed a similar improvement in performance even when type features are added or the setting is changed to **typed+gen** or with Model 2.

Setting	Accuracy
Independent	87.17
+ Shared	**88.01**

Table 7: Comparing the effect of feature sharing across prepositions. We see that having a shared representation that goes across prepositions improves accuracy of relation prediction ($p < 0.01$).

Different argument candidate generators Our second ablation study looks at the effect of the various argument candidate generators. Recall that in addition to the dependency governor and object, our models also use the previous word, the previous noun, adjective and verb as governor candidates and the next noun as object candidate. We refer to the candidates generated by the parser as *Parser only* and the others as *Heuristics only*. Table 8 compares the performance of these two argument candidate generators against the full set using Model 1 in both the **typed** and **typed+gen** settings.

We see that the heuristics give a better accuracy than the parser based system. This is because the heuristics often contain the governor/object predicted by the dependency. This is not always the case, though, because using all generators gives a slightly better performing system (not statistically significant). In the overall system, we retain the dependency parser as one of the generators in order to capture long-range governor/object candidates that may not be in the set selected by the heuristics.

	Feature sets	
Generator	**typed**	**typed+gen**
Parser only	87.12	87.12
Heuristics only	87.63	88.84
All	88.01	89.12

Table 8: The performance of different argument candidate generators. We see that considering a larger set of candidate generators gives a big accuracy improvement.

6 Discussion

There are two key differences between Model 1 and 2. First, the former predicts only the relation label, while the latter predicts the entire structure. Table 9 shows example predictions of Model 2 for relation label and WordNet argument types. These examples show how the argument types can be thought of as an explanation for the choice of relation label.

Input	Relation	Hypernyms	
		governor	object
died of pneumonia	`Cause`	experience	disease
suffered from flu	`Cause`	experience	disease
recovered from flu	`StartState`	change	disease

Table 9: Example predictions according to Model 2. The hypernyms column shows a representative of the synset chosen for the WordNet types. We see that in the combination of *experience* and *disease* suggests the relation `Cause` while the *change* and *disease* indicate the relation `StartState`.

The main difference between the two models is in the treatment of the unlabeled (or latent) parts of the structure (namely, the arguments and the types) during training and inference. During training, for

each example, Model 1 aggregates features from all governors and objects even if they are possibly irrelevant, which may lead to a much bigger model in terms of the number of active weights. On the other hand, for Model 2, Algorithm 1 uses the *single* highest scoring prediction of the latent variables, according to the current parameters, to refine the parameters. Indeed, in our experiments, we observed that the number of non-zero weights in the weight vector of Model 2 is much smaller than that of Model 1. For instance, in the **typed** setting, the weight vector for Model 1 had 2.57 million elements while that for Model 2 had only 1.0 million weights. Similarly, for the **typed+gen** setting, Model 1 had 5.41 million non-zero elements in the weight vector while Model 2 had only 2.21 million non-zero elements.

The learning algorithm itself is a generalization of the latent structural SVM of (Yu and Joachims, 2009). By setting α to zero, we get the latent structure SVM. However, we found via cross-validation that this is not the best setting of the parameter. A theoretical understanding of the sparsity of weights learned by the algorithm and a study of its convergence properties is an avenue of future research.

7 Conclusion

We addressed the problem of modeling semantic relations expressed by prepositions. We approached this task by defining a set of preposition relations that combine preposition senses across prepositions. Doing so allowed us to leverage existing annotated preposition sense data to induce a corpus for preposition labels. We modeled preposition relations in terms of its arguments, namely the governor and object of the preposition, *and* the semantic types of the arguments. Using a generalization of the latent structural SVM, we trained a relation, argument and type predictor using only annotated relation labels. This allowed us to get an accuracy of 89.43% on relation prediction. By employing joint inference with a preposition sense predictor, we further improved the relation accuracy to 90.23%.

Acknowledgments

The authors wish to thank Martha Palmer, Nathan Schneider, the anonymous reviewers and the editor for their valuable feedback. The authors gratefully acknowledge the support of the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181. This material is also based on research sponsored by DARPA under agreement number FA8750-13-2-0008. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA, AFRL or the U.S. Government.

References

E. Agirre, T. Baldwin, and D. Martinez. 2008. Improving parsing and PP attachment performance with sense information. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 317–325, Columbus, USA.

T. Baldwin, V. Kordoni, and A. Villavicencio. 2009. Prepositions in applications: A survey and introduction to the special issue. *Computational Linguistics*, 35(2):119–149.

M. Chang, D. Goldwasser, D. Roth, and V. Srikumar. 2010a. Discriminative learning over constrained latent representations. In *Proceedings of the Annual Meeting of the North American Association of Computational Linguistics (NAACL)*, pages 429–437, Los Angeles, USA.

M. Chang, V. Srikumar, D. Goldwasser, and D. Roth. 2010b. Structured output learning with indirect supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 199–206, Haifa, Israel.

J. Clarke, V. Srikumar, M. Sammons, and D. Roth. 2012. An NLP Curator (or: How I Learned to Stop Worrying and Love NLP Pipelines). In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, pages 3276–3283, Istanbul, Turkey.

J. Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46.

D. Dahlmeier, H. T. Ng, and T. Schultz. 2009. Joint learning of preposition senses and semantic roles of prepositional phrases. In *Proceedings of the Conference on Empirical Methods for Natural Language Processing (EMNLP)*, pages 450–458, Singapore.

D. Das, N. Schneider, D. Chen, and N. Smith. 2010. Probabilistic frame-semantic parsing. In *Proceedings of Human Language Technologies: The 2010 Annual*

Conference of the North American Chapter of the Association for Computational Linguistics, pages 948–956, Los Angeles, USA.

C. Fillmore, C. Johnson, and M. Petruck. 2003. Background to FrameNet. *International Journal of Lexicography*, 16(3):235–250.

D. Gildea and D. Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Y. Goldberg and M. Elhadad. 2010. An efficient algorithm for easy-first non-directional dependency parsing. In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 742–750, Los Angeles, USA.

D. Hovy, S. Tratz, and E. Hovy. 2010. What's in a preposition? dimensions of sense disambiguation for an interesting word class. In *Coling 2010: Posters*, pages 454–462, Beijing, China.

D. Lin. 1998. Automatic retrieval and clustering of similar words. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 768–774, Montreal, Canada.

K. Litkowski and O. Hargraves. 2005. The Preposition Project. In *ACL-SIGSEM Workshop on the Linguistic Dimensions of Prepositions and their Use in Computational Linguistics Formalisms and Applications*, pages 171–179, Colchester, UK.

K. Litkowski and O. Hargraves. 2007. SemEval-2007 Task 06: Word-Sense Disambiguation of Prepositions. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 24–29, Prague, Czech Republic.

K. Litkowski. 2012. Proposed Next Steps for The Preposition Project. Technical Report 12-01, CL Research.

A. Meyers, R. Reeves, C. Macleod, R. Szekely, V. Zielinska, B. Young, and R. Grishman. 2004. The NomBank project: An interim report. In *HLT-NAACL 2004 Workshop: Frontiers in Corpus Annotation*, pages 24–31, Boston, USA.

T. O'Hara and J. Wiebe. 2009. Exploiting semantic role resources for preposition disambiguation. *Computational Linguistics*, 35(2):151–184.

M. Palmer, D. Gildea, and P. Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics*, 31(1):71–106.

M. Palmer, D. Gildea, and N. Xue. 2010. *Semantic Role Labeling*, volume 3. Morgan & Claypool Publishers.

V. Punyakanok, D. Roth, and W. Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2).

D. Roth and W. Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Annual Conference on Computational Natural Language Learning (CoNLL)*, pages 1–8, Boston, USA.

D. Roth and W. Yih. 2007. Global inference for entity and relation identification via a linear programming formulation. *Introduction to Statistical Relational Learning*.

V. Srikumar and D. Roth. 2011. A joint model for extended semantic role labeling. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Edinburgh, Scotland.

S. Tratz and D. Hovy. 2009. Disambiguation of preposition sense using linguistically motivated features. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Student Research Workshop and Doctoral Consortium*, pages 96–100, Boulder, USA.

S. Tratz. 2011. *Semantically-enriched Parsing for Natural Language Understanding*. Ph.D. thesis, University of Southern California.

I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. 2004. Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 104–111, Banff, Canada.

P. Ye and T. Baldwin. 2006. Semantic role labeling of prepositional phrases. *ACM Transactions on Asian Language Information Processing (TALIP)*, 5(3):228–244.

P. Ye and T. Baldwin. 2007. MELB-YB: Preposition sense disambiguation using rich semantic features. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 241–244, Prague, Czech Republic.

C. Yu and T. Joachims. 2009. Learning structural SVMs with latent variables. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 1–8, Montreal, Canada.

B. Zapirain, E. Agirre, L. Màrquez, and M. Surdeanu. 2012. Selectional preferences for semantic role classification. *Computational Linguistics*, pages 1–33.