Improving NLP through Marginalization of Hidden Syntactic Structure

Jason Naradowsky, Sebastian Riedel, and **David A. Smith**
Department of Computer Science
University of Massachusetts Amherst
Amherst, MA, 01003, U.S.A.
{narad, riedel, dasmith}@cs.umass.edu

Abstract

Many NLP tasks make predictions that are inherently coupled to syntactic relations, but for many languages the resources required to provide such syntactic annotations are unavailable. For others it is unclear exactly how much of the syntactic annotations can be effectively leveraged with current models, and what structures in the syntactic trees are most relevant to the current task.

We propose a novel method which avoids the need for any syntactically annotated data when predicting a related NLP task. Our method couples latent syntactic representations, constrained to form valid dependency graphs or constituency parses, with the prediction task via specialized factors in a Markov random field. At both training and test time we marginalize over this hidden structure, learning the optimal latent representations for the problem. Results show that this approach provides significant gains over a syntactically uninformed baseline, outperforming models that observe syntax on an English relation extraction task, and performing comparably to them in semantic role labeling.

1 Introduction

Many NLP tasks are inherently tied to syntax, and state-of-the-art solutions to these tasks often rely on syntactic annotations as either a source for useful features (Zhang et al., 2006, path features in relation extraction) or as a scaffolding upon which a more narrow, specialized classification can occur (as often done in semantic role labeling). This decoupling of the end task from its intermediate representation is sometimes known as the **two-stage** approach (Chang et al., 2010) and comes with several drawbacks. Most notably this decomposition prohibits the learning method from utilizing the labels from the end task when predicting the intermediate representation, a structure which must have some correlation to the end task to provide any benefit.

Relying on intermediate representations that are specifically syntactic in nature introduces its own unique set of problems. Large amounts of syntactically annotated data is difficult to obtain, costly to produce, and often tied to a particular domain that may vary greatly from that of the desired end task. Additionally, current systems often utilize only a small amount of the annotation for any particular task. For instance, performing named entity recognition (NER) jointly with constituent parsing has been shown to improve performance on both tasks, but the only aspect of the syntax which is leveraged by the NER component is the location of noun phrases (Finkel and Manning, 2009). By instead discovering a latent representation jointly with the end task we address all of these concerns, alleviating the need for any syntactic annotations, while simultaneously attempting to learn a latent syntax relevant to both the particular domain and structure of the end task.

We phrase the joint model as factor graph and marginalize over the hidden structure of the intermediate representation at both training and test time, to optimize performance on the end task. Inference is done via loopy belief propagation, making this framework trivially extensible to most graph structures. Computation over latent syntactic rep-

resentations is made tractable with the use of special combinatorial factors which implement unlabeled variants of common dynamic-programming parsing algorithms, constraining the hidden representation to realize valid dependency graphs or constituency trees.

We apply this strategy to two common NLP tasks, coupling a model for the end task prediction with latent and general syntactic representations via specialized logical factors which learn associations between latent and observed structure. In comparisons with identical models which observe "gold" syntactic annotations, derived from off-the-shelf parsers or provided with the corpora, we find that our hidden marginalization method is comparable in both tasks and almost every language tested, sometimes significantly outperforming models which observe the true syntax.

The following sections serves as a preliminary, introducing an inventory of factors and variables for constructing factor graph representations of syntactically-coupled NLP tasks. Section 3 explores the benefits of this method on relation extraction (RE), where we compare the use dependency and constituency structure as latent representations. We then turn to a more established semantic role labeling (SRL) task (§4) where we evaluate across a wide range of languages.

2 Latent Pseudo-Syntactic Structure

The models presented in this paper are phrased in terms of variables in an undirected graphical model, Markov random field. More specifically, we implement the model as a factor graph, a bipartite graph composed of factors and variables in which we can efficiently compute the marginal beliefs of any variable set with the sum-product algorithm for cyclic graphs, *loopy belief propagation*,. We now introduce the basic variable and factor components used throughout the paper.

2.1 Latent Dependency Structure

Dependency grammar is a lexically-oriented syntactic formalism in which syntactic relationships are expressed as dependencies between individual words. Each non-root word specifies another as its head, provided that the resulting structure forms a valid directed graph, ie. there are no cycles in the graph. Due to the flexibility of this representation it is often used to describe free-word-order languages, and increasingly preferred in NLP for more language-in-use scenarios. A dependency graph can be modeled with the following nodes, as first proposed by Smith and Eisner (2008):

- Let $\{Link(i,j) : 0 \leq i \leq j \leq n, n \neq j\}$ be $O(n^2)$ boolean variables corresponding to the possible links in a dependency parse. $L_{i,j}$ = true implies that there is a dependency from parent i to child j.
- Let $\{LINK(i,j) : 0 \leq i \leq j \leq n, n \neq j\}$ be $O(n^2)$ unary factors, each paired with a corresponding $Link(i,j)$ variable and expressing the independent belief that $Link(i,j)$ = true.

2.2 Latent Constituency Structure

Alternatively we can describe the more structured constituency formalism by setting up a representation over span variables:

- Let $\{Span(i,j) : 0 \leq i < j \leq n\}$ be $O(n^2)$ boolean variables such that $Span(i,j) =$ true iff there is a bracket spanning i to j [1].
- Let $\{\textsc{Span}(i,j) : 0 \leq i < j \leq n\}$ be $O(n^2)$ unary factors, each attached to the corresponding $Span(i,j)$ variable. These factors score the independent suitability of each span to appear in an unlabeled constituency tree.

All boolean variables presented in this paper will be paired to unary factors in this manner, which we will omit in future descriptions. This encompasses the necessary *representational* structure for both syntactic formalisms, but nothing introduced up to this point guarantees that either of these representations will form a valid tree or DAG.

2.3 Combinatorial Factors

Naively constraining these latent representations through the introduction of many interconnected ternary factors is possible, but would likely be computationally intractable. However, as observed

[1] In practice, we do not need to include variables for spans of width 1 or n, since they will always be true.

in Smith and Eisner (2008), we can encapsulate common dynamic programming algorithms within special-purpose factors to efficiently globally constrain variable configurations. Since the outgoing messages from such factors to a variable can be computed from the factor's posterior beliefs about that variable, there is no difficulty in exchanging beliefs between these special-purpose factors and the rest of the graph, and inference can proceed using the standard sum-product or max-product belief propagation. Here we present two combinatorial factors that provide efficient ways of constraining the model to fit common syntactic frameworks.

- Let CKYTREE be a global combinatorial factor, as used in previous work in efficient parsing (Naradowsky and Smith, 2012), attached to all the $Span(i, j)$ variables. This factor contributes a factor of 1 to the model's score iff the span variables collectively form a legal, binary bracketing and a factor of 0 otherwise. It enforces, therefore, a hard constraint on the variables, computing beliefs via an unlabeled variant of the inside-outside algorithm.

- Let DEP-TREE be a global combinatorial factor, as presented in Smith and Eisner (2008), which attaches to all $Link(i, j)$ variables and similarly contributes a factor of 1 iff the configuration of $Link$ variables forms a valid projective dependency graph. A graph is projective if its edges do not cross.

2.4 Marginal MAP Inference

It is straightforward to train these latent variable models to maximize the marginal probability of their outputs, conditioning on their inputs, and marginalizing out the latent syntactic variables. To compute feature expectations, we can use marginal inference techniques such as sampling and sum-product belief propagation to compute marginal probabilities.

A knottier problem arises when we want to find the best assignment to the variables of interest while marginalizing out "nuisance" latent variables. This is the problem of **marginal MAP** inference—sometimes known as consensus decoding—which has been shown to be NP-hard and without a polynomial time approximation scheme (Sima'an, 1996; Casacuberta and Higuera, 2000). In the NLP community, these inference problems often arise when dealing with *spurious ambiguity* where multiple derivations can lead to the same derived structure. In tree substitution grammars, for instance, there may be many ways of combining elementary trees to produce the same output tree; in machine translation, many different elementary phrases or elementary tree pairs might produce the same output string. For syntactic parsing, Goodman (1996) proposed a variational method for summing out spurious ambiguity that was equivalent to minimum Bayes risk decoding (Goel and Byrne, 2000; Kumar and Byrne, 2004) with a constituent-recall loss function. For MT, May and Knight (2006) proposed methods for determinizing tree automata to reduce ambiguity, and Li et al. (2009) proposed a variational method based on n-gram loss functions. More recently, Liu and Ihler (2011) analyzed message-passing algorithms for marginal MAP.

In this paper, we adopt a simple minimum Bayes risk decoding scheme. First, we perform sum-product belief propagation on the full factor graph. Then, we maximize the expected accuracy of the variables of interest, subject to any hard constraints on them (such as mutual exclusion among labels). In some cases with complex combinatorial constraints, this simple MBR scheme has proved more effective than exact decoding over all variables (Auli and Lopez, 2011).

3 Relation Extraction

Performing a syntax-based NLP task in most real-world scenarios requires that the incoming data first be parsed using a pre-trained parsing model. For some tasks, like relation extraction, many data sets lack syntactic annotation and these circumstances persist even into the training phase. In this section we explore such scenarios and contrast the use of parser-provided syntactic annotation to marginalizing over latent representations of constituency or dependency syntax. We show the hidden syntactic models are not just competitive with these "oracle" models, but in some configurations can actually outperform them.

Relation extraction is the task of identifying semantic relations between sets of entities in text (as

illustrated in Fig. 1b), and a good proving ground for latent syntactic methods for two reasons. First, because entities share a semantic relationship, under most linguistic analyses these entities will also share some syntactic relation. Indeed, syntactic features have long been an extremely useful source of information for relation extraction systems (Culotta and Sorensen, 2004; Mintz et al., 2009). Secondly, relation extraction has been a common task for pioneering efforts in processing data mined from the internet, and otherwise noisy or out-of-domain data. In particular, large noisily-annotated data sets have been generated by leveraging freely available knowledge bases such as Freebase (Bollacker et al., 2008; Mintz et al., 2009). Such data sets have been utilized successfully for relation extraction from the web (Bunescu and Mooney, 2007).

3.1 Model

We present a simple model for representing relational structure, with the only variables present being a set of boolean-valued variables representing an undirected dependency between two entities, and an additional set of boolean label variables representing the type label of the relation.

- Let $\{Rel(i, j : 0 \leq i < j \leq n\}$ be $O(n^2)$ boolean variables such that $Rel(i, j) = \textsf{true}$ iff there is a relation spanning i to j.

- Let $\{Rel\text{-}Label(i, j, \lambda) : \lambda \in L$, and $0 \leq i < j \leq n\}$ be $O(|L|n^2)$ boolean variables such that $Rel\text{-}Label(i, j, \lambda) = \textsf{true}$ iff there is a relation spanning i to j with relation type λ.

- Let $\{\text{AtMost1}(i, j) : 0 \leq i < j \leq n\}$ be $O(n^2)$ factors, each coordinating the set L of possible nonterminal variables to the Rel variable at each i, j tuple, allowing a $Rel\text{-}Label$ variable to be $\textsf{true}$ iff all other label variables are $\textsf{false}$ and $Rel(i, j) = \textsf{true}$.

Here the $Rel(i, j)$ and $Rel\text{-}Label(i, j)$ variables simply express the representation of the problem, while the AtMost1 factors are logical constraints ensuring that only one label will apply to a particular relation.

3.2 Coordination Factors

An important contribution of this work is the introduction of a flexible, general framework for connecting the latent and observable partitions of the model. We accomplish this through the use of two additional factors, each expressing the same basic logic, which learn when to coordinate and when to ignore correlations between the latent syntax and the end task. While here we specify binary and ternary versions of these factors, they also generalize to higher dimensions.

- Let $\{\text{D-Connect}(i, j, k) : 0 \leq i < j \leq n; 0 \leq k \leq n\}$ be $O(n^3)$ factors coordinating any number of dependency syntax $Link(i, j)$ variables with representational variables on the end task, multiplying in 1 to the model score unless all variables are on, in which case it multiplies a connective potential ϕ derived from its features. Thus it functions logically as a soft NAND factor. In this ternary formulation k represents a hidden dependency head or pivot which is shared between two syntactic dependencies anchored at the indices of the entities in the relation (as illustrated in Fig. 1).

- Let $\{\text{C-Connect}(i, j) : 0 \leq i < j \leq n\}$ be $O(n^2)$ factors coordinating syntactic $Span(i, j)$ and relation arc $Rel(i, j)$, identically to D-Connect but with a 1-to-1 mapping. Intuitively the joint model might learn $\phi > 1$, i.e., constituency spans and task prediction relations are more likely to be coterminous.

The difficulty in working with latent dependency syntax is that we posit that the RE variables do not share a 1-to-1 mapping with variables in the hidden representation. We expect instead, according to linguistic intuition, that a relation between entities at position i and j in the sentence should have corresponding syntactic dependencies but that they are likely to realize this by sharing the same head word (as depicted in Fig.1), a word whose identity should help label the relation. Therefore we introduce a special coordination factor, D-Connect as a ternary factor to capture the relationship between pairs of latent syntactic variables and a single relation variable, pivoting on the same unknown head word.

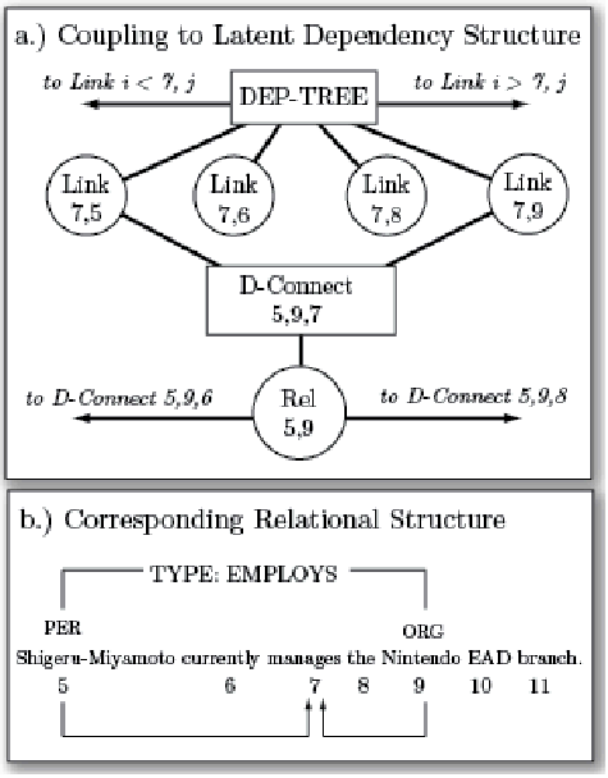

Figure 1: Latent Dependency coupling for the RE task. The D-CONNECT factor expresses ternary connection relations because the shared head word of the proposed relation is unknown. As is convention, variables are represented by circles, factors by rectangles.

We introduce six model scenarios.

- **Baseline**, simply the arc-factored model consisting only of *Rel* and corresponding *Label* variables for each entity. Features on the relation factors, which are common to all model configurations, are combinations of lexical information (i.e., the words that form the entity, the pos-tags of the entities, etc.) as well as the distance between the relation. This is a lightweight model and generally does not attempt to exhaustively leverage all possible proven sources of useful features (Zhou et al., 2005) towards a higher absolute score, but rather to serve as a point of comparison to the models which rely on syntactic information.

- **Baseline-Ent**, a variant of **Baseline** with additional features which include combinations of mention type, entity type, and entity sub-type.

- **Oracle D-Parse**, in which we also instantiate a full set of latent dependency syntax variables, and connect them to the baseline model using D-CONNECT factors. Syntax variables are clamped to their true values.

- **Oracle C-Parse**, the constituency syntax analogue of **Oracle D-Parse**.

- **Hidden D-Parse**, which is an extension of **Oracle D-Parse** in which we connect all syntax variables to a DEP-TREE factor, syntax variables are unobserved, and are learned jointly with the end task. The features for latent syntax are a subset of those used in dependency parsing (McDonald et al., 2005).

- **Hidden C-Parse**, the constituency syntax analogue of **Hidden D-Parse**. The feature set is similar but bigrams are taken over the words defining the constituent span, rather than the words defining the head/modifier relation.

Coordination factor features for the syntactically-informed models are particularly important. This became evident in initial experiments where the baseline was often able to outperform the hidden syntactic model. However, inclusion of entity and mention label features into the connection factors provides the model with greater reasoning over when to coordinate or ignore the relation predictions with the underlying syntax. These are a proper subset of the **Baseline-Ent** features.

3.3 Data

We evaluate these models using the 2005 Automatic Content Extraction (ACE) data set (Walker, 2006), using the English (dual-annotated) and Chinese (solely annotator #1 data set) sections. Each corpus is annotated with entity mentions—tagged as PER, ORG, LOC, or MISC—and, where applicable, what type of relation exists between them (e.g., coarse: PHYS; fine: Located). But like most corpora available for the task, the burden of acquiring corresponding syntactic annotation is left to the researcher. In this situation it is common to turn to existing pre-trained parsing models.

We generate our data by first splitting the raw text paragraphs into sentences. Chinese sentences

ACE Results												
	English						Chinese					
	Unlabeled			Labeled			Unlabeled			Labeled		
Model	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Baseline	85.4	57.0	68.4	83.0	55.3	66.4	42.9	26.8	33.0	42.6	21.3	28.4
Baseline-Ent	87.2	65.4	74.8	85.8	64.4	73.6	55.2	31.1	39.8	51.2	29.4	37.4
Oracle D-Parse	89.3	67.4	76.8	89.3	66.2	75.4	60.0	32.6	42.2	58.1	31.3	40.7
Hidden D-Parse	87.8	69.8	77.7	85.3	67.8	75.6	48.0	32.0	38.4	47.2	30.0	36.7
Oracle C-Parse	89.1	68.7	77.6	87.5	67.5	76.2	66.8	37.8	**48.3**	63.8	37.0	**46.8**
Hidden C-Parse	90.5	69.9	**78.9**	88.8	68.6	**77.4**	56.3	32.3	41.0	53.4	31.6	39.7

Table 1: Relation Extraction Results. Models using hidden constituency syntax provide significant gains over the syntactically-uniformed baseline model in both languages, but the advantages of the latent syntax were mitigated on the smaller Chinese data set.

are also tokenized according to Penn Chinese Treebank standards (Xue et al., 2005). The sentences are then tagged and parsed using the Stanford CoreNLP tools, using the standard pre-trained models for tagging (Toutanvoa and Manning, 2000), and the factored parsing model of Klein and Manning (2002). The distributed grammar is trained on a variety of sources, including the standard Wallstreet Journal corpus, but also biomedical, translation, and questions. We then apply entity and relation annotations noisily to the data, collapsing multi-word entities into one term. We filter out sentences with fewer than two entities (and are thus incapable of containing relations) and sentences with more than 40 words (to keep the parses more reliable). This yields 6966 sentences for English data, but unfortunately only 747 sentences for the Chinese. Nine of every ten sentences comprise the training set, with every tenth sentence reserved for test.

3.4 Results

We train all models using 20 iterations of stochastic gradient descent, each with a maximum of 10 BP iterations (though in practice we find convergence to often occur much earlier). The results are presented in Table 1, showing precision, recall, and F-measure for both labeled and unlabeled prediction. For English, not only is the hidden marginalization method a suitable replacement for the syntactic trees provided by pre-trained, state-of-the-art models, but in both configurations we find that inducing an optimal hidden structure is preferable to the parser-produced annotations. On Chinese, where the data set is atypically small, we still observe improved performance over the baseline in the constituency-based model though it is not able to match the observed syntax model.

Despite the intuition that both entities occupy roles as modifiers of the same verb, we find that the **Hidden D-Parse** model often fails to recover the correct latent structure, and that even when successful dependency parses are observed, the head word is often not uniquely indicative of the relation type (as *known* is not strongly correlated with the relation type EMPLOYS in the phrase: *Shigeru Miyamoto, best known for his work at the video game company Nintendo*). Hence when it comes to relation extraction, at least on our relatively small data sets, we find the simplest approach to latent syntactic structure is the best.

We now turn to the task of semantic role labeling to evaluate this method on a more established hand-annotated data set, and a more varied set of languages.

4 Semantic Role Labeling

The task of semantic role labeling (SRL) aims to detect and label the semantic relationships between particular words, most commonly verbs (referred to in the domain as *predicates*), and their *arguments* (Meza-Ruiz and Riedel, 2009).

In a manner similar to RE, there is a strong correlation between the presence of an SRL relation and there existing an underlying syntactic dependency, though this is not always expressed as directly as a 1-to-1 correspondence. This has historically motivated a reliance on syntactic annotation, and some of the most successful methods have simply applied

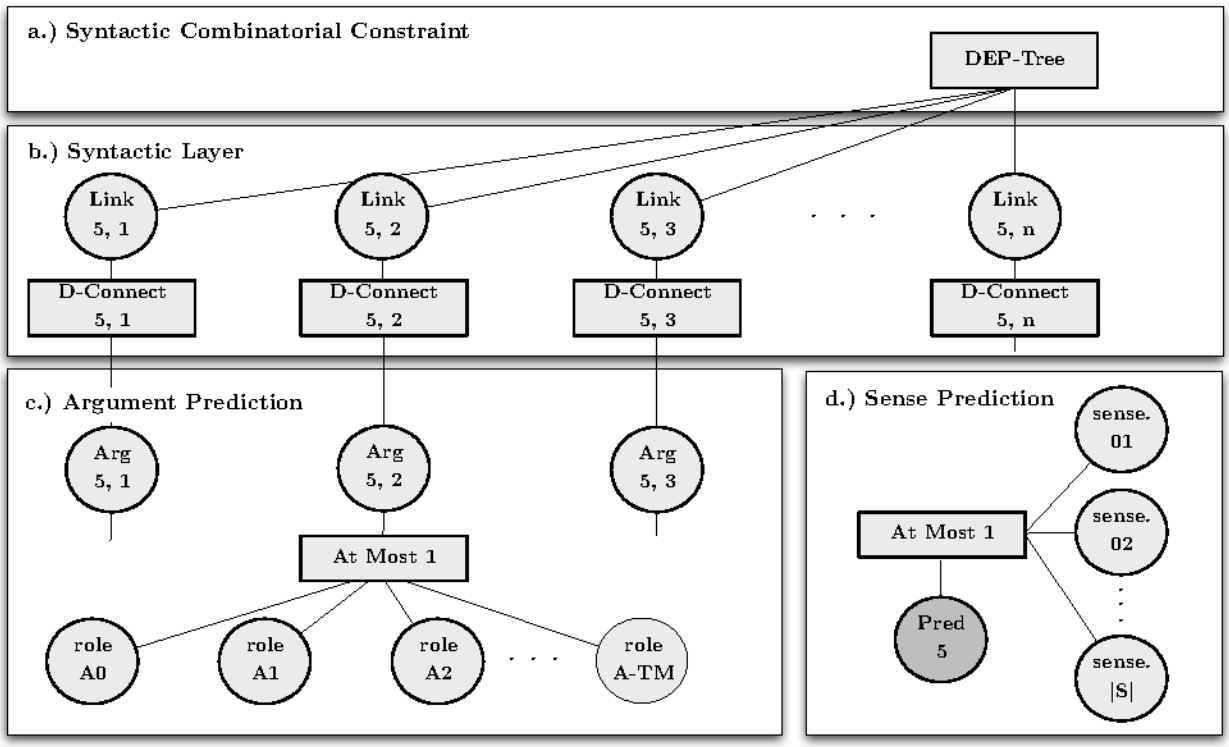

Figure 2: A tiered graphic representing the three different SRL model configurations. The baseline system is described in the bottom (c & d), the separate panels highlighting the independent predictions of this model: sense labels are assigned in an entirely separate process from argument prediction. Pruning in the model takes place primarily in this tier, since we observe true predicates we only instantiate over these indices. The middle tier (b.) illustrates the syntactic representation layer, and the connective factors between syntax and SRL. In the observed syntax model the *Link* variables are clamped to their correct values, with no need for a factor to coordinate them to form a valid tree. Finally, the hidden model comprises all layers, including a combinatorial syntactic constraint (a.) over syntactic variables. In this scenario all labels in (b.) are hidden at both training and test time.

feature-rich classifiers to the parsed trees. Related work has recognized the large annotation burden the task demands, but aimed to keep the syntactic annotations and induce semantic roles (Lang and Lapata, 2010). In this section we will take the opposite approach, disregarding the syntactic annotations which we argue are more costly to acquire, as they require more formal linguistic training to produce.

4.1 Model

We present a simple, flexible model for SRL in which sense predictions are made independently of the rest of the model, and argument predictions are made independently of each other. The model structure is composed as depicted in Fig. 2.

- Let $\{Arg(i,j) : 0 \leq i < j \leq n\}$ be $O(n^2)$ boolean variables such that $Arg(i,j) =$ true iff predicate i takes token j as an argument.
- Let $\{Role(i,j,\lambda) : \lambda \in L$, and $0 \leq i < j \leq n\}$ be $O(|L|n^2)$ boolean variables such that $Role(i,j,\lambda) =$ true iff $Arg(i,j)$ is true and takes the role label λ.
- Let $\{Sense(i,\sigma) : \sigma \in S$, and $0 \leq i \leq n\}$ be $O(|S|n)$ boolean variables such that $Sense(i,\sigma) =$ true iff predicate i has sense σ.

4.1.1 Features

At the coarsest level both the SRL and RE models are specifying binary predictions between a pair of indices in the sentence, and a set of labels for each dependency that happens to be true. Similarly we use almost identical features in SRL as we did in

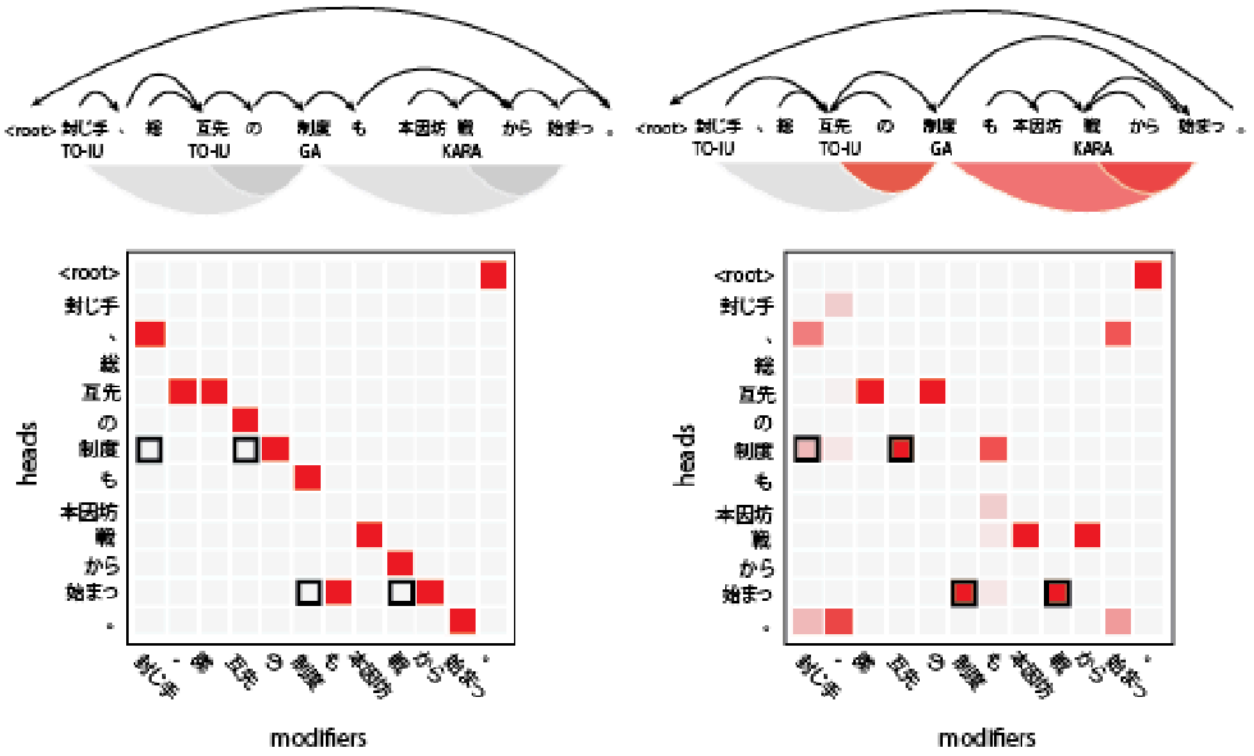

Figure 3: Examining the learned hidden representation for SRL. In this example the syntactic dependency arcs derived from gold standard syntactic annotations (*left*) are entirely disjoint from the correct predicate/arguments pairs (shown in the heatmaps by the squares outlined in black), and the observed syntax model fails to recover any of the correct predictions. In contrast, the hidden model structure (*right*) learns a representation that closely parallels the desired end task predictions, helping it recover three of the four correct SRL predictions (shaded arcs: red corresponds to a correct prediction, with true labels GA, KARA, etc.), and providing some evidence towards the fourth. The dependency tree corresponding to the hidden structure is derived by edge-factored decoding: dependency variables whose beliefs > 0.5 are classified as true (though some arcs not relevant to the SRL predictions are omitted for clarity).

RE, with the sole exception that we incorporate the observable lemma and morphological features into bigrams on predicate/argument pairs. For sense prediction we rely only on unigram features taken in a close ($w = 2$) window of the target predicate.

For the coordinating factors we use subsets of combinations of word, part-of-speech, and capitalization features taken between head and argument, and concatenate these with the distance and direction between the predicate and argument. We do not find the performance of the system to be as sensitive to which features are present in the coordinating factors as we did in the RE task.

4.2 Data

We evaluate our SRL model using the data set developed for the CoNLL 2009 shared task competition (Hajič et al., 2009), which features seven languages and provides an ideal opportunity to measure the ability of the hidden structure to generalize across languages of disparate origin and varied characteristics. It also provides the opportunity to observe a variety of different annotation styles and biases, some of which our model was able to uncover as ill-suited to common models for the task. The data itself provides word, lemma, part-of-speech, and morphological feature information, along with gold dependency parses. Words which denote predicates are identified, and their (train time) arguments are provided. They are also annotated with a sense label for each predicate, which is scored as an additional SRL dependency. Thus the task involves predicting for each predicate a set of argument dependencies and the sense label associated with that predicate.

Data	Model	Unlabeled			Labeled			CoNLL 2009 F1		
		P	R	F1	P	R	F1	MAX.	MEAN	MED.
Catalan	Baseline	92.20	62.43	74.48	73.80	58.76	65.43	80.3	71.0	74.1
	Oracle Syn.	98.48	96.17	**97.31**	70.42	68.78	**69.59**			
	Hidden Syn.	95.21	92.84	94.01	68.86	67.15	67.99			
Chinese	Baseline	72.48	64.82	68.44	65.97	59.00	62.29	78.6	72.2	70.4
	Oracle Syn.	98.57	78.98	**87.69**	87.64	70.22	**77.97**			
	Hidden Syn.	90.79	79.09	84.53	81.97	71.40	76.32			
Czech	Baseline	97.73	56.50	71.61	84.80	48.80	61.84	85.4	72.4	71.7
	Oracle Syn.	98.62	81.25	89.09	92.94	68.25	**74.84**			
	Hidden Syn.	92.39	89.35	**90.85**	74.41	71.96	73.16			
English	Baseline	92.46	71.56	80.68	84.56	65.45	73.78	85.6	75.6	72.1
	Oracle Syn.	96.75	82.25	**88.91**	85.48	72.67	**78.55**			
	Hidden Syn.	95.06	79.06	86.32	83.82	69.72	76.12			
German	Baseline	93.49	44.24	60.06	75.00	35.49	48.18	79.7	68.1	67.8
	Oracle Syn.	95.18	79.11	86.41	73.24	60.87	66.49			
	Hidden Syn.	91.92	86.26	**89.00**	69.47	65.19	**67.26**			
Japanese	Baseline	91.64	43.36	58.87	80.41	38.05	51.66	78.2	62.7	72.0
	Oracle Syn.	93.84	48.15	63.64	90.06	46.21	61.08			
	Hidden Syn.	90.88	73.47	**81.25**	73.42	59.36	**65.65**			
Spanish	Baseline	82.90	39.47	53.48	67.64	32.21	43.64	80.5	70.4	73.4
	Oracle Syn.	98.96	94.19	**96.52**	70.68	67.27	**68.93**			
	Hidden Syn.	96.15	90.53	93.25	68.81	64.79	66.74			

Table 2: SRL Results. The hidden model excels on the unlabeled prediction results, often besting the scores obtained using the parses distributed with the CoNLL data sets. These gains did not always translate to the labeled task where poor sense prediction hindered absolute performance.

4.3 Results

We evaluate across a set of model configurations analogous to before. All experiments used 30 iterations of SGD with a Gaussian prior, and a max 10 iterations of BP to compute the marginals for each example. In comparison to the CoNLL competition entries (Table 2, rightmost columns) our syntactically-informed models generally fall in the middle of the rankings. This is not surprising given the independent predictions of the model and the very general, language universal assumptions we have made in the model structure and feature sets. However, in terms of gauging the usefulness of the hidden syntactic marginalization method the results are extremely compelling, with only marginal differences between the performance of the observed-syntax model, especially relative to the baseline.

And despite the simplicity of the model, we still manage to perform at state-of-the-art levels in a few instances, sometimes outperforming most of the competition entries without observing any syntax. The performance on Chinese is an example of this, with our system outperforming all but the best system, and the hidden syntactic model only slightly behind.

Abstracting away from the performance comparisons against other systems, the unlabeled results are the more revealing evidence for the use of hidden syntactic structure. Here the average hidden model score (88.89) almost outperforms the observed syntax model (90.22, and vs. 66.80 baseline), mostly due to the large margins on the unlabeled Japanese scores. The strong independence between sense prediction and argument prediction hinders performance on the labeled task, but on all languages we find an extremely significant improvement exploiting hidden syntactic structure in comparison to the baseline system—the hidden model recovers more than 92% of the gap between the baseline and the observed syntax model. It is also interesting to note that in the shared task competition the two languages which systems lost the most performance between their parsing F1 and their SRL F1 were Japanese and German. As illustrated in Fig. 3, the corre-

spondence between syntax and SRL are extremely, and systematically, poor. In this example our hidden structure model was able to assign strong beliefs to the latent syntactic variables which correspond to the correct predicate/argument pairs, allowing it to correctly identify three of the four SRL arguments when the joint model failed to recover one.

5 Related Work

This work is perhaps mostly closely related to the Learning over Constrained Latent Representations (LCLR) framework of Chang et al. (2010). Their abstract problem formulation is identical: both paradigms seek to couple the end task to an intermediate representation which is not accessible to the learning algorithm. However much of the intent, scale, and methodology is different. LCLR aims to provide a flexible latent structure for increasing the representational power of the model in a useful way, and is demonstrated on tasks and domains where data availability is not a key concern. In contrast, while our hidden structure models may outperform their observed syntax counterparts, our focus is as much on alleviating the burden of procuring large amounts of syntactic annotation as it is about increasing the expressiveness of the model. To that end we constrain a more sophisticated latent representation and couple it to highly structured output predictions, opposed to binary classification problems. In methodology, we perform the more computationally intensive marginalization operation instead of maximizing.

Marginalization of hidden structure is also fundamental to other work, and featured most prominently in generative Bayesian latent variable models (Teh et al., 2006). Our approach is trained discriminatively, affording the use of very rich feature sets and the prediction of partial structures without needing to specify a full derivation. Similar approaches have been used in more linear latent variable CRF-based models (McCallum et al., 2005), but these must only marginalize only over hidden states of a much more compact representation. Naively extending this to tree-based constraints would often be computationally inefficient, and we avoid intractability through the encapsulation of much of the dynamic programming machinery into specialized factors. Moreover, using loopy belief propagation means that the inference method is not closely coupled to the task structure, and need not change when applying this method to other types of graphs.

6 Conclusion

We have presented a novel method of coupling syntactically-oriented NLP tasks to combinatorially-constrained hidden syntactic representations, and have shown that marginalizing over this latent representation not only provides significant improvements over syntactically-uninformed baselines, but occasionally improves performance when compared to systems which observe syntax. On the task of relation extraction we find that a constituency representation provides the most improvement over the baseline, while in the SRL domain our model is extremely competitive with the best reported results on Chinese, and outperforms the model using the provided parses on German and Japanese.

We believe this method delivers very promising results in our presented tasks, opening the door to new lines of research examining what types of constraints and what configurations of hidden structure are most beneficial for particular tasks and languages. Moreover, we present one type of coordinating factor, as both D-CONNECT and C-CONNECT logically express a soft NAND function, but more sophisticated coupling schemes are another natural direction to pursue. Finally, we use sum-product variant of belief propagation inference, but more specialized inference schemes may show additional benefits.

Acknowledgements

We would like to thank Andrea Gesmundo for help in procuring sections of the CoNLL 2009 shared task data. This work was supported in part by the Center for Intelligent Information Retrieval and in part by Army prime contract number W911NF-07-1-0216 and University of Pennsylvania subaward number 103-548106. The University of Massachusetts also gratefully acknowledges the support of 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 authors and do not necessarily reflect the view of the DARPA, AFRL, or the US government.

References

Michael Auli and Adam Lopez. 2011. A comparison of loopy belief propagation and dual decomposition for integrated CCG supertagging and parsing. In *ACL*, pages 470–480.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *SIGMOD*, pages 1247–1250, New York, NY, USA. ACM.

Razvan C. Bunescu and Raymond J. Mooney. 2007. Learning to extract relations from the web using minimal supervision. In *ACL*.

Francisco Casacuberta and Colin De La Higuera. 2000. Computational complexity of problems on probabilistic grammars and transducers. In *ICGI*, pages 15–24.

M. Chang, D. Goldwasser, D. Roth, and V. Srikumar. 2010. Discriminative learning over constrained latent representations. In *NAACL*.

Aron Culotta and Jeffery Sorensen. 2004. Dependency tree kernels for relation extraction. In *ACL*, Barcelona, Spain.

Jenny Rose Finkel and Christopher D. Manning. 2009. Joint parsing and named entity recognition. In *NAACL*, pages 326–334.

Vaibbhava Goel and William J. Byrne. 2000. Minimum Bayes risk automatic speech recognition. *Computer Speech and Language*, 14(2):115–135.

Joshua T. Goodman. 1996. Parsing algorithms and metrics. In *ACL*, pages 177–183.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *CoNLL: Shared Task*, pages 1–18.

Dan Klein and Chris Manning. 2002. Fast exact inference with a factored model for natural language processing. In *NIPS*.

Shankar Kumar and William Byrne. 2004. Minimum Bayes-risk decoding for statistical machine translation. In *HLT-NAACL*, pages 169–176.

Joel Lang and Mirella Lapata. 2010. Unsupervised induction of semantic roles. In *HLT-NAACL*, pages 939–947.

Zhifei Li, Jason Eisner, and Sanjeev Khudanpur. 2009. Variational decoding for statistical machine translation. In *ACL*, pages 593–601.

Qiang Liu and Alexander Ihler. 2011. Variational algorithms for marginal MAP. In *UAI*, pages 453–462.

Jonathan May and Kevin Knight. 2006. A better n-best list: Practical determinization of weighted finite tree automata. In *HLT-NAACL*, pages 351–358.

Andrew McCallum, Kedar Bellare, and Fernando C. N. Pereira. 2005. A conditional random field for discriminatively-trained finite-state string edit distance. In *UAI*, pages 388–395.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *HLT-EMNLP*, pages 523–530.

Ivan Meza-Ruiz and Sebastian Riedel. 2009. Jointly identifying predicates, arguments and senses using Markov logic. In *NAACL*.

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

Jason Naradowsky and David A. Smith. 2012. Combinatorial constraints for constituency parsing in graphical novels. Technical report, University of Massachusetts Amherst.

Khalil Sima'an. 1996. Computational complexity of probabilistic disambiguation by means of tree-grammars. In *COLING*, pages 1175–1180.

David A. Smith and Jason Eisner. 2008. Dependency parsing by belief propagation. In *EMNLP*, pages 145–156.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Kristina Toutanvoa and Christopher D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *EMNLP*, pages 63–70.

Christopher Walker. 2006. Ace 2005 multilingual training corpus. number ldc2006t06. In *Linguistic Data Consortium*.

Nianwen Xue, Fei Xia, Fu-Dong Chiou, and Martha Palmer. 2005. The penn chinese treebank: Phrase structure annotation of a large corpus. In *Natural Language Engineering*, pages 207–238.

Min Zhang, Jie Zhang, and Jian Su. 2006. Exploring syntactic features for relation extraction using a convolution tree kernel. In *NAACL*, pages 288–295.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *ACL*, pages 427–434.