Low-Resource Semantic Role Labeling

Matthew R. Gormley[1] **Margaret Mitchell**[2] **Benjamin Van Durme**[1] **Mark Dredze**[1]
[1]Human Language Technology Center of Excellence
Johns Hopkins University, Baltimore, MD 21211
[2]Microsoft Research
Redmond, WA 98052
mrg@cs.jhu.edu | memitc@microsoft.com | vandurme@cs.jhu.edu | mdredze@cs.jhu.edu

Abstract

We explore the extent to which high-resource manual annotations such as treebanks are necessary for the task of semantic role labeling (SRL). We examine how performance changes without syntactic supervision, comparing both *joint* and *pipelined* methods to induce latent syntax. This work highlights a new application of unsupervised grammar induction and demonstrates several approaches to SRL in the absence of supervised syntax. Our best models obtain competitive results in the high-resource setting and state-of-the-art results in the low resource setting, reaching 72.48% F1 averaged across languages. We release our code for this work along with a larger toolkit for specifying arbitrary graphical structure.[1]

1 Introduction

The goal of semantic role labeling (SRL) is to identify predicates and arguments and label their semantic contribution in a sentence. Such labeling defines *who* did *what* to *whom*, *when*, *where* and *how*. For example, in the sentence "The kids ran the marathon", *ran* assigns a role to *kids* to denote that they are the runners; and a role to *marathon* to denote that it is the race course.

Models for SRL have increasingly come to rely on an array of NLP tools (e.g., parsers, lemmatizers) in order to obtain state-of-the-art results (Björkelund et al., 2009; Zhao et al., 2009). Each tool is typically trained on hand-annotated data, thus placing SRL at the end of a very high-resource NLP pipeline. However, richly annotated data such as that provided in parsing treebanks is expensive to produce, and may be tied to specific domains (e.g., newswire). Many languages do not have such supervised resources (*low-resource languages*), which makes exploring SRL cross-linguistically difficult.

The problem of SRL for low-resource languages is an important one to solve, as solutions pave the way for a wide range of applications: Accurate identification of the semantic roles of entities is a critical step for any application sensitive to semantics, from information retrieval to machine translation to question answering.

In this work, we explore models that minimize the need for high-resource supervision. We examine approaches in a **joint** setting where we marginalize over latent syntax to find the optimal semantic role assignment; and a **pipeline** setting where we first induce an unsupervised grammar. We find that the joint approach is a viable alternative for making reasonable semantic role predictions, outperforming the pipeline models. These models can be effectively trained with access to only SRL annotations, and mark a state-of-the-art contribution for low-resource SRL.

To better understand the effect of the low-resource grammars and features used in these models, we further include comparisons with (1) models that use higher-resource versions of the same features; (2) state-of-the-art high resource models; and (3) previous work on low-resource grammar induction. In sum, this paper makes several experimental and modeling contributions, summarized below.

Experimental contributions:

- Comparison of pipeline and joint models for SRL.
- Subtractive experiments that consider the removal of supervised data.
- Analysis of the induced grammars in unsupervised, distantly-supervised, and joint training settings.

[1]http://www.cs.jhu.edu/~mrg/software/

Modeling contributions:

- Simpler joint CRF for syntactic and semantic dependency parsing than previously reported.
- New application of unsupervised grammar induction: low-resource SRL.
- Constrained grammar induction using SRL for distant-supervision.
- Use of Brown clusters in place of POS tags for low-resource SRL.

The pipeline models are introduced in § 3.1 and jointly-trained models for syntactic and semantic dependencies (similar in form to Naradowsky et al. (2012)) are introduced in § 3.2. In the pipeline models, we develop a novel approach to unsupervised grammar induction and explore performance using SRL as distant supervision. The joint models use a non-loopy conditional random field (CRF) with a global factor constraining latent syntactic edge variables to form a tree. Efficient exact marginal inference is possible by embedding a dynamic programming algorithm within belief propagation as in Smith and Eisner (2008).

Even at the expense of no dependency path features, the joint models best pipeline-trained models for state-of-the-art performance in the low-resource setting (§ 4.4). When the models have access to observed syntactic trees, they achieve near state-of-the-art accuracy in the high-resource setting on some languages (§ 4.3).

Examining the learning curve of the joint and pipeline models in two languages demonstrates that a small number of labeled SRL examples may be essential for good end-task performance, but that the choice of a good model for grammar induction has an even greater impact.

2 Related Work

Our work builds upon research in both semantic role labeling and unsupervised grammar induction (Klein and Manning, 2004; Spitkovsky et al., 2010a). Previous related approaches to semantic role labeling include joint classification of semantic arguments (Toutanova et al., 2005; Johansson and Nugues, 2008), latent syntax induction (Boxwell et al., 2011; Naradowsky et al., 2012), and feature engineering for SRL (Zhao et al., 2009; Björkelund et al., 2009).

Toutanova et al. (2005) introduced one of the first joint approaches for SRL and demonstrated that a model that scores the full predicate-argument structure of a parse tree could lead to significant error reduction over independent classifiers for each predicate-argument relation.

Johansson and Nugues (2008) and Lluís et al. (2013) extend this idea by coupling predictions of a dependency parser with predictions from a semantic role labeler. In the model from Johansson and Nugues (2008), the outputs from an SRL pipeline are reranked based on the full predicate-argument structure that they form. The candidate set of syntactic-semantic structures is reranked using the probability of the syntactic tree and semantic structure. Lluís et al. (2013) use a joint arc-factored model that predicts full syntactic paths along with predicate-argument structures via dual decomposition.

Boxwell et al. (2011) and Naradowsky et al. (2012) observe that syntax may be treated as latent when a treebank is not available. Boxwell et al. (2011) describe a method for training a semantic role labeler by extracting features from a packed CCG parse chart, where the parse weights are given by a simple ruleset. Naradowsky et al. (2012) marginalize over latent syntactic dependency parses.

Both Boxwell et al. (2011) and Naradowsky et al. (2012) suggest methods for SRL without supervised syntax, however, their features come largely from supervised resources. Even in their lowest resource setting, Boxwell et al. (2011) require an oracle CCG tag dictionary extracted from a treebank. Naradowsky et al. (2012) limit their exploration to a small set of basic features, and included high-resource supervision in the form of lemmas, POS tags, and morphology available from the CoNLL 2009 data.

There has not yet been a comparison of techniques for SRL that do not rely on a syntactic treebank, and no exploration of probabilistic models for unsupervised grammar induction within an SRL pipeline that we have been able to find.

Related work for the unsupervised learning of dependency structures separately from semantic roles primarily comes from Klein and Manning (2004), who introduced the Dependency Model with Valence (DMV). This is a robust generative model that uses a head-outward process over word classes, where heads generate arguments.

Spitkovsky et al. (2010a) show that Viterbi (hard) EM training of the DMV with simple uniform initialization of the model parameters yields higher accuracy models than standard soft-EM

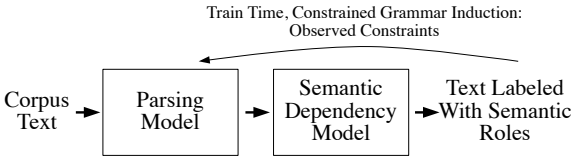

Figure 1: Pipeline approach to SRL. In this simple pipeline, the first stage syntactically parses the corpus, and the second stage predicts semantic predicate-argument structure for each sentence using the labels of the first stage as features. In our *low-resource* pipelines, we assume that the syntactic parser is given no labeled parses—however, it may optionally utilize the semantic parses as distant supervision. Our experiments also consider 'longer' pipelines that include earlier stages: a morphological analyzer, POS tagger, lemmatizer.

training. In Viterbi EM, the E-step finds the maximum likelihood corpus parse given the current model parameters. The M-step then finds the maximum likelihood parameters given the corpus parse. We utilize this approach to produce unsupervised syntactic features for the SRL task.

Grammar induction work has further demonstrated that distant supervision in the form of ACE-style relations (Naseem and Barzilay, 2011) or HTML markup (Spitkovsky et al., 2010b) can lead to considerable gains. Recent work in fully unsupervised dependency parsing has supplanted these methods with even higher accuracies (Spitkovsky et al., 2013) by arranging optimizers into networks that suggest informed restarts based on previously identified local optima. We do not reimplement these approaches within the SRL pipeline here, but provide comparison of these methods against our grammar induction approach in isolation in § 4.5.

In both pipeline and joint models, we use features adapted from state-of-the-art approaches to SRL. This includes Zhao et al. (2009) features, who use feature templates from combinations of word properties, syntactic positions including head and children, and semantic properties; and features from Björkelund et al. (2009), who utilize features on syntactic siblings and the dependency path concatenated with the direction of each edge. Features are described further in § 3.3.

3 Approaches

We consider an array of models, varying:

1. Pipeline vs. joint training (Figures 1 and 2)
2. Types of supervision
3. The objective function at the level of syntax

3.1 Unsupervised Syntax in the Pipeline

Typical SRL systems are trained following a pipeline where the first component is trained on supervised data, and each subsequent component is trained using the 1-best output of the previous components. A typical pipeline consists of a POS tagger, dependency parser, and semantic role labeler. In this section, we introduce pipelines that remove the need for a supervised tagger and parser by training in an unsupervised and distantly supervised fashion.

Brown Clusters We use fully unsupervised Brown clusters (Brown et al., 1992) in place of POS tags. Brown clusters have been used to good effect for various NLP tasks such as named entity recognition (Miller et al., 2004) and dependency parsing (Koo et al., 2008; Spitkovsky et al., 2011). The clusters are formed by a greedy hierachical clustering algorithm that finds an assignment of words to classes by maximizing the likelihood of the training data under a latent-class bigram model. Each word type is assigned to a fine-grained cluster at a leaf of the hierarchy of clusters. Each cluster can be uniquely identified by the path from the root cluster to that leaf. Representing this path as a bit-string (with 1 indicating a left and 0 indicating a right child) allows a simple coarsening of the clusters by truncating the bit-strings. We train 1000 Brown clusters for each of the CoNLL-2009 languages on Wikipedia text.[2]

Unsupervised Grammar Induction Our first method for grammar induction is *fully unsupervised* Viterbi EM training of the Dependency Model with Valence (DMV) (Klein and Manning, 2004), with uniform initialization of the model parameters. We define the DMV such that it generates sequences of word classes: either POS tags or Brown clusters as in Spitkovsky et al. (2011). The DMV is a simple generative model for projective dependency trees. Children are generated recursively for each node. Conditioned on the parent class, the direction (right or left), and the current valence (first child or not), a coin is flipped to decide whether to generate another child; the distribution over child classes is conditioned on only the parent class and direction.

[2]The Wikipedia text was tokenized for Polyglot (Al-Rfou' et al., 2013): `http://bit.ly/embeddings`

Constrained Grammar Induction Our second method, which we will refer to as DMV+C, induces grammar in a *distantly supervised* fashion by using a constrained parser in the E-step of Viterbi EM. Since the parser is part of a pipeline, we constrain it to respect the downstream SRL annotations during training. At test time, the parser is unconstrained.

Dependency-based semantic role labeling can be described as a simple structured prediction problem: the predicted structure is a labeled directed graph, where nodes correspond to words in the sentence. Each directed edge indicates that there is a predicate-argument relationship between the two words; the parent is the predicate and the child the argument. The label on the edge indicates the type of semantic relationship. Unlike syntactic dependency parsing, the graph is not required to be a tree, nor even a connected graph. Self-loops and crossing arcs are permitted.

The constrained *syntactic* DMV parser treats the semantic graph as observed, and constrains the syntactic parent to be chosen from one of the semantic parents, if there are any. In some cases, imposing this constraint would not permit *any* projective dependency parses—in this case, we ignore the semantic constraint for that sentence. We parse with the CKY algorithm (Younger, 1967; Aho and Ullman, 1972) by utilizing a PCFG corresponding to the DMV (Cohn et al., 2010). Each chart cell allows only non-terminals compatible with the constrained sets. This can be viewed as a variation of Pereira and Schabes (1992).

Semantic Dependency Model As described above, semantic role labeling can be cast as a structured prediction problem where the structure is a labeled semantic dependency graph. We define a conditional random field (CRF) (Lafferty et al., 2001) for this task. Because each word in a sentence may be in a semantic relationship with any other word (including itself), a sentence of length n has n^2 possible edges. We define a single L+1-ary variable for each edge, whose value can be any of L semantic labels or a special label indicating there is no predicate-argument relationship between the two words. In this way, we jointly perform *identification* (determining whether a semantic relationship exists) and *classification* (determining the semantic label). This use of an L+1-ary variable is in contrast to the model of Naradowsky et al. (2012), which used a more complex set of binary variables and required a constraint factor permitting AT-MOST-ONE. We include one unary factor for each variable.

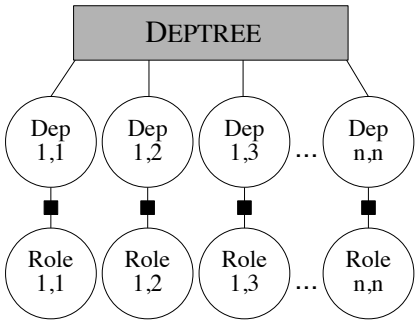

Figure 2: Factor graph for the joint syntactic/semantic dependency parsing model.

We optionally include additional variables that perform word sense disambiguation for each predicate. Each has a unary factor and is completely disconnected from the semantic edge (similar to Naradowsky et al. (2012)). These variables range over all the predicate senses observed in the training data for the *lemma* of that predicate.

3.2 Joint Syntactic and Semantic Parsing Model

In Section 3.1, we introduced pipeline-trained models for SRL, which used grammar induction to predict unlabeled syntactic parses. In this section, we define a simple model for joint syntactic and semantic dependency parsing.

This model extends the CRF model in Section 3.1 to include the projective syntactic dependency parse for a sentence. This is done by including an additional n^2 binary variables that indicate whether or not a directed syntactic dependency edge exists between a pair of words in the sentence. Unlike the semantic dependencies, these syntactic variables must be coupled so that they produce a projective dependency parse; this requires an additional global constraint factor to ensure that this is the case (Smith and Eisner, 2008). The constraint factor touches all n^2 syntactic-edge variables, and multiplies in 1.0 if they form a projective dependency parse, and 0.0 otherwise. We couple each syntactic edge variable to its semantic edge variable with a binary factor. Figure 2 shows the factor graph for this joint model.

Note that our factor graph does not contain any loops, thereby permitting efficient exact marginal inference just as in Naradowsky et al. (2012). We

Property		Possible values
1	word form	all word forms
2	lower case word form	all lower-case forms
3	5-char word form prefixes	all 5-char form prefixes
4	capitalization	*True, False*
5	top-800 word form	top-800 word forms
6	brown cluster	*000, 1100, 010110001, ...*
7	brown cluster, length 5	length 5 prefixes of brown clusters
8	lemma	all word lemmas
9	POS tag	*NNP, CD, JJ, DT, ...*
10	morphological features (different across languages)	Gender, Case, Number, ...
11	dependency label	*SBJ, NMOD, LOC, ...*
12	edge direction	*Up, Down*

Table 1: Word and edge properties in templates.

i, i-1, i+1	noFarChildren(w_i)	linePath(w_p, w_c)
parent(w_i)	rightNearSib(w_i)	depPath(w_p, w_c)
allChildren(w_i)	leftNearSib(w_i)	depPath(w_p, w_{lca})
rightNearChild(w_i)	firstVSupp(w_i)	depPath(w_c, w_{lca})
rightFarChild(w_i)	lastVSupp(w_i)	depPath(w_{lca}, w_{root})
leftNearChild(w_i)	firstNSupp(w_i)	
leftFarChild(w_i)	lastNSupp(w_i)	

Table 2: Word positions used in templates. Based on current word position (i), positions related to current word w_i, possible parent, child (w_p, w_c), lowest common ancestor between parent/child (w_{lca}), and syntactic root (w_{root}).

train our CRF models by maximizing conditional log-likelihood using stochastic gradient descent with an adaptive learning rate (AdaGrad) (Duchi et al., 2011) over mini-batches.

The unary and binary factors are defined with exponential family potentials. In the next section, we consider binary features of the observations (the sentence and labels from previous pipeline stages) which are conjoined with the state of the variables in the factor.

3.3 Features for CRF Models

Our feature design stems from two key ideas. First, for SRL, it has been observed that feature bigrams (the concatenation of simple features such as a predicate's POS tag and an argument's word) are important for state-of-the-art (Zhao et al., 2009; Björkelund et al., 2009). Second, for syntactic dependency parsing, combining Brown cluster features with word forms or POS tags yields high accuracy even with little training data (Koo et al., 2008).

We create binary indicator features for each model using feature templates. Our feature template definitions build from those used by the top performing systems in the CoNLL-2009 Shared Task, Zhao et al. (2009) and Björkelund et al. (2009) and from features in syntactic dependency parsing (McDonald et al., 2005; Koo et al., 2008).

Template	Possible values
relative position	*before, after, on*
distance, continuity	$\mathbb{Z}^+$
binned distance	$>$ 2, 5, 10, 20, 30, or 40
geneological relationship	*parent, child, ancestor, descendant*
path-grams	*the_NN_went*

Table 3: Additional standalone templates.

Template Creation Feature templates are defined over triples of $\langle$property, positions, order$\rangle$. **Properties**, listed in Table 1, are extracted from word **positions** within the sentence, shown in Table 2. Single positions for a word w_i include its syntactic parent, its leftmost farthest child (leftFarChild), its rightmost nearest sibling (rightNearSib), etc. Following Zhao et al. (2009), we include the notion of verb and noun supports and sections of the dependency path. Also following Zhao et al. (2009), properties from a set of positions can be put together in three possible **orders**: as the given sequence, as a sorted list of unique strings, and removing all duplicated neighbored strings. We consider both template unigrams and bigrams, combining two templates in sequence.

Additional templates we include are the relative position (Björkelund et al., 2009), geneological relationship, distance (Zhao et al., 2009), and binned distance (Koo et al., 2008) between two words in the path. From Lluís et al. (2013), we use $1, 2, 3$-gram path features of words/POS tags (path-grams), and the number of non-consecutive token pairs in a predicate-argument path (continuity).

3.4 Feature Selection

Constructing all feature template unigrams and bigrams would yield an unwieldy number of features. We therefore determine the top N template bigrams for a dataset and factor a according to an information gain measure (Martins et al., 2011):

$$IG_{a,m} = \sum_{f \in T_m} \sum_{x_a} p(f, x_a) \log_2 \frac{p(f, x_a)}{p(f)p(x_a)}$$

where T_m is the mth feature template, f is a particular instantiation of that template, and x_a is an assignment to the variables in factor a. The probabilities are empirical estimates computed from the training data. This is simply the mutual information of the feature template instantiation with the variable assignment.

This filtering approach was treated as a simple baseline in Martins et al. (2011) to contrast with increasingly popular gradient based regularization approaches. Unlike the gradient based ap-

proaches, this filtering approach easily scales to many features since we can decompose the memory usage over feature templates.

As an additional speedup, we reduce the dimensionality of our feature space to 1 million for each clique using a common trick referred to as *feature hashing* (Weinberger et al., 2009): we map each feature instantiation to an integer using a hash function[3] modulo the desired dimentionality.

4 Experiments

We are interested in the effects of varied supervision using pipeline and joint training for SRL. To compare to prior work (i.e., submissions to the CoNLL-2009 Shared Task), we also consider the joint task of semantic role labeling *and* predicate sense disambiguation. Our experiments are subtractive, beginning with all supervision available and then successively removing (a) dependency syntax, (b) morphological features, (c) POS tags, and (d) lemmas. Dependency syntax is the most expensive and difficult to obtain of these various forms of supervision. We explore the importance of both the labels and structure, and what quantity of supervision is useful.

4.1 Data

The CoNLL-2009 Shared Task (Hajič et al., 2009) dataset contains POS tags, lemmas, morphological features, syntactic dependencies, predicate senses, and semantic roles annotations for 7 languages: Catalan, Chinese, Czech, English, German, Japanese,[4] Spanish. The CoNLL-2005 and -2008 Shared Task datasets provide English SRL annotation, and for cross dataset comparability we consider only verbal predicates (more details in § 4.4). To compare with prior approaches that use semantic supervision for grammar induction, we utilize Section 23 of the WSJ portion of the Penn Treebank (Marcus et al., 1993).

4.2 Feature Template Sets

Our primary feature set $\mathbf{IG}_C$ consists of 127 template unigrams that emphasize coarse properties (i.e., properties 7, 9, and 11 in Table 1). We also explore the 31 template unigrams[5] $\mathbf{IG}_B$ described by Björkelund et al. (2009). Each of IG_C and IG_B also include 32 template bigrams selected by information gain on 1000 sentences—we select a different set of template bigrams for each dataset.

We compare against the language-specific feature sets detailed in the literature on high-resource top-performing SRL systems: From Björkelund et al. (2009), these are feature sets for German, English, Spanish and Chinese, obtained by weeks of forward selection ($\mathbf{B}_{de,en,es,zh}$); and from Zhao et al. (2009), these are features for Catalan $\mathbf{Z}_{ca}$.[6]

4.3 High-resource SRL

We first compare our models trained as a pipeline, using all available supervision (syntax, morphology, POS tags, lemmas) from the CoNLL-2009 data. Table 4(a) shows the results of our model with gold syntax and a richer feature set than that of Naradowsky et al. (2012), which only looked at whether a syntactic dependency edge was present. This highlights an important advantage of the pipeline trained model: the features can consider any part of the syntax (e.g., arbitrary subtrees), whereas the joint model is limited to those features over which it can efficiently marginalize (e.g., short dependency paths). This holds true even in the pipeline setting where no syntactic supervision is available.

Table 4(b) contrasts our high-resource results for the task of SRL and sense disambiguation with the top systems in the CoNLL-2009 Shared Task, giving further insight into the performance of the simple information gain feature selection technique. With supervised syntax, our simple information gain feature selection technique (§ 3.4) performs admirably. However, the original unigram Björkelund features ($\text{B}_{de,en,es,zh}$), which were tuned for a high-resource model, obtain higher F1 than our information gain set using the same features in unigram and bigram templates (IG_B). This suggests that further work on feature selection may improve the results. We find that IG_B obtain *higher* F1 than the original Björkelund feature sets ($\text{B}_{de,en,es,zh}$) in the low-resource pipeline setting with constrained grammar induction (DMV+C).

[3] To reduce hash collisions, We use MurmurHash v3 `https://code.google.com/p/smhasher`.

[4] We do not report results on Japanese as that data was only made freely available to researchers that competed in CoNLL 2009.

[5] Because we do not include a binary factor between predicate sense and semantic role, we do not include sense as a feature for argument prediction.

[6] This covers all CoNLL languages but Czech, where feature sets were not made publicly available in either work. In Czech, we disallowed template bigrams involving path-grams.

	SRL Approach	Feature Set	Dep. Parser	Avg.	ca	cs	de	en	es	zh
(a)	Pipeline	IG_C	Gold	**84.98**	84.97	**87.65**	79.14	**86.54**	84.22	**87.35**
	Pipeline	IG_B	Gold	84.74	**85.15**	86.64	**79.50**	85.77	**84.40**	86.95
	Naradowsky et al. (2012)		Gold	72.73	69.59	74.84	66.49	78.55	68.93	77.97
(b)	Björkelund et al. (2009)		Supervised	**81.55**	80.01	**85.41**	**79.71**	**85.63**	79.91	**78.60**
	Zhao et al. (2009)		Supervised	80.85	**80.32**	85.19	75.99	85.44	**80.46**	77.72
	Pipeline	IG_C	Supervised	78.03	76.24	83.34	74.19	81.96	76.12	76.35
	Pipeline	Z_{ca}	Supervised	*77.62	77.62	—	—	—	—	—
	Pipeline	$B_{de,en,es,zh}$	Supervised	*76.49	—	—	72.17	81.15	76.65	75.99
	Pipeline	IG_B	Supervised	75.68	74.59	81.61	69.08	78.86	74.51	75.44
(c)	Joint	IG_C	Marginalized	**72.48**	71.35	**81.03**	65.15	**76.16**	71.03	70.14
	Joint	IG_B	Marginalized	72.40	**71.55**	80.04	64.80	75.57	**71.21**	71.21
	Naradowsky et al. (2012)		Marginalized	71.27	67.99	73.16	**67.26**	76.12	66.74	**76.32**
	Pipeline	IG_C	DMV+C (bc)	70.08	68.21	79.63	62.25	73.81	68.73	67.86
	Pipeline	Z_{ca}	DMV+C (bc)	*69.67	69.67	—	—	—	—	—
	Pipeline	IG_C	DMV (bc)	69.26	68.04	79.58	58.47	74.78	68.36	66.35
	Pipeline	IG_B	DMV (bc)	66.81	63.31	77.38	59.91	72.02	65.96	62.28
	Pipeline	IG_B	DMV+C (bc)	65.61	61.89	77.48	58.97	69.11	63.31	62.92
	Pipeline	$B_{de,en,es,zh}$	DMV+C (bc)	*63.06	—	—	57.75	68.32	63.70	62.45

Table 4: Test F1 for SRL and sense disambiguation on CoNLL'09 in high-resource and low-resource settings: we study (a) gold syntax, (b) supervised syntax, and (c) unsupervised syntax. Results are ranked by F1 with bold numbers indicating the best F1 for a language and level of supervision.
*Indicates partial averages for the language-specific feature sets (Z_{ca} and $B_{de,en,es,zh}$), for which we show results only on the languages for which the sets were publicly available.

train \ test		*2008 heads*	*2005 spans*	*2005 spans (oracle tree)*
☑ PRY'08	*2005 spans*	84.32	79.44	
☐ B'11 (tdc)		—	71.5	
☐ B'11 (td)		—	65.0	
☑ JN'08	*2008 heads*	85.93	79.90	
☐ Joint, IG_C		72.9	35.0	72.0
☐ Joint, IG_B		67.3	37.8	67.1

Table 5: F1 for SRL approaches (without sense disambiguation) in matched and mismatched train/test settings for CoNLL 2005 span and 2008 head supervision. We contrast low-resource (☐) and high-resource settings (☑), where latter uses a treebank. See § 4.4 for caveats to this comparison.

4.4 Low-Resource SRL

CoNLL-2009 Table 4(c) includes results for our low-resource approaches and Naradowsky et al. (2012) on predicting semantic roles as well as sense. In the low-resource setting of the CoNLL-2009 Shared task without syntactic supervision, our joint model (Joint) with marginalized syntax obtains state-of-the-art results with features IG_C described in § 4.2. This model outperforms prior work (Naradowsky et al., 2012) and our pipeline model (Pipeline) with contrained (DMV+C) and unconstrained grammar induction (DMV) trained on brown clusters (bc).

In the low-resource setting, training and decoding times for the pipeline and joint methods are similar as computation time tends to be dominated by feature extraction.

These results begin to answer a key research question in this work: The joint models outperform the pipeline models in the low-resource setting. This holds even when using the same feature selection process. Further, the best-performing low-resource features found in this work are those based on coarse feature templates and selected by information gain. Templates for these features generalize well to the high-resource setting. However, analysis of the induced grammars in the pipeline setting suggests that the book is not closed on the issue. We return to this in § 4.5.

CoNLL-2008, -2005 To finish out comparisons with state-of-the-art SRL, we contrast our approach with that of Boxwell et al. (2011), who evaluate on SRL in isolation (without sense disambiguation, as in CoNLL-2009). They report results on Prop-CCGbank (Boxwell and White, 2008), which uses the same training/testing splits as the CoNLL-2005 Shared Task. Their results are therefore loosely[7] comparable to results on the CoNLL-2005 dataset, which we can compare here.

There is an additional complication in comparing SRL approaches directly: The CoNLL-2005 dataset defines arguments as *spans* instead of

[7]The comparison is imperfect for two reasons: first, the CCGBank contains only 99.44% of the original PTB sentences (Hockenmaier and Steedman, 2007); second, because PropBank was annotated over CFGs, after converting to CCG only 99.977% of the argument spans were exact matches (Boxwell and White, 2008). However, this comparison was adopted by Boxwell et al. (2011), so we use it here.

heads, which runs counter to our head-based syntactic representation. This creates a mismatched train/test scenario: we must train our model to predict argument *heads*, but then test on our models ability to predict argument *spans*.[8] We therefore train our models on the CoNLL-2008 argument heads,[9] and post-process and convert from heads to spans using the conversion algorithm available from Johansson and Nugues (2008).[10] The heads are either from an MBR tree or an oracle tree. This gives Boxwell et al. (2011) the advantage, since our syntactic dependency parses are optimized to pick out semantic argument heads, not spans.

Table 5 presents our results. Boxwell et al. (2011) (B'11) uses additional supervision in the form of a CCG tag dictionary derived from supervised data with (tdc) and without (tc) a cutoff. Our model does very poorly on the '05 span-based evaluation because the constituent bracketing of the marginalized trees are inaccurate. This is elucidated by instead evaluating on the oracle spans, where our F1 scores are higher than Boxwell et al. (2011). We also contrast with relavant high-resource methods with span/head conversions from Johansson and Nugues (2008): Punyakanok et al. (2008) (PRY'08) and Johansson and Nugues (2008) (JN'08).

Subtractive Study In our subsequent experiments, we study the effectiveness of our models as the available supervision is decreased. We incrementally remove dependency syntax, morphological features, POS tags, then lemmas. For these experiments, we utilize the coarse-grained feature set (IG_C), which includes Brown clusters.

Across languages, we find the largest drop in F1 when we remove POS tags; and we find a gain in F1 when we remove lemmas. This indicates that lemmas, which are a high-resource annotation, may not provide a significant benefit for this task. The effect of removing morphological features is different across languages, with little change in performance for Catalan and Spanish, but a drop in performance for German. This may reflect a difference between the languages, or may reflect the difference between the annotation of the languages: both the Catalan and Spanish data originated from the Ancora project,[11] while the German data came from another source.

Rem	#FT	ca	de	es
–	127+32	74.46	72.62	74.23
Dep	40+32	67.43	64.24	67.18
Mor	30+32	67.84	59.78	66.94
POS	23+32	64.40	54.68	62.71
Lem	21+32	64.85	54.89	63.80

Table 6: Subtractive experiments. Each row contains the F1 for SRL only (without sense disambiguation) where the supervision type of that row and all above it have been removed. Removed supervision types (Rem) are: syntactic dependencies (*Dep*), morphology (*Mor*), POS tags (*POS*), and lemmas (*Lem*). #FT indicates the number of feature templates used (unigrams+bigrams).

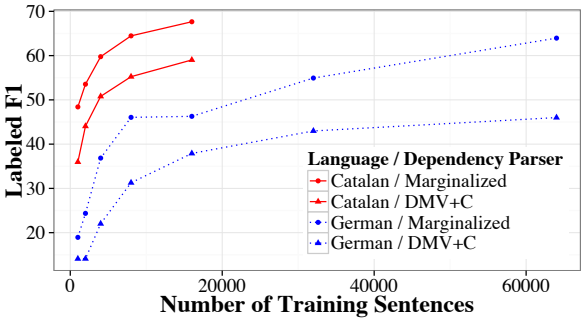

Figure 3: Learning curve for semantic dependency supervision in Catalan and German. F1 of SRL only (without sense disambiguation) shown as the number of training sentences is increased.

Figure 3 contains the learning curve for SRL supervision in our lowest resource setting for two example languages, Catalan and German. This shows how F1 of SRL changes as we adjust the number of training examples. We find that the joint training approach to grammar induction yields consistently higher SRL performance than its distantly supervised counterpart.

4.5 Analysis of Grammar Induction

Table 7 shows grammar induction accuracy in low-resource settings. We find that the gap between the supervised parser and the unsupervised methods is quite large, despite the reasonable accuracy both methods achieve for the SRL end task.

[8] We were unable to obtain the system output of Boxwell et al. (2011) in order to convert their spans to dependencies and evaluate the other mismatched train/test setting.

[9] CoNLL-2005, -2008, and -2009 were derived from PropBank and share the same source text; -2008 and -2009 use argument heads.

[10] Specifically, we use their Algorithm 2, which produces the span dominated by each argument, with special handling of the case when the argument head dominates that of the predicate. Also following Johansson and Nugues (2008), we recover the '05 sentences missing from the '08 evaluation set.

[11] http://clic.ub.edu/corpus/ancora

Dependency Parser	Avg.	ca	cs	de	en	es	zh
Supervised*	87.1	89.4	85.3	89.6	88.4	89.2	80.7
DMV (pos)	30.2	45.3	22.7	20.9	32.9	41.9	17.2
DMV (bc)	22.1	18.8	32.8	19.6	22.4	20.5	18.6
DMV+C (pos)	37.5	50.2	34.9	21.5	36.9	49.8	32.0
DMV+C (bc)	40.2	46.3	37.5	28.7	40.6	50.4	37.5
Marginal, IG_C	43.8	50.3	45.8	27.2	44.2	46.3	48.5
Marginal, IG_B	50.2	52.4	43.4	41.3	52.6	55.2	56.2

Table 7: Unlabeled directed dependency accuracy on CoNLL'09 test set in low-resource settings. DMV models are trained on either POS tags (pos) or Brown clusters (bc). *Indicates the supervised parser outputs provided by the CoNLL'09 Shared Task.

	WSJ^{∞}	Distant Supervision
SAJM'10	44.8	none
SAJ'13	**64.4**	none
SJA'10	50.4	HTML
NB'11	59.4	ACE05
DMV (bc)	24.8	none
DMV+C (bc)	44.8	SRL
Marginalized, IG_C	48.8	SRL
Marginalized, IG_B	58.9	SRL

Table 8: Comparison of grammar induction approaches. We contrast the DMV trained with Viterbi EM+uniform initialization (DMV), our constrained DMV (DMV+C), and our model's MBR decoding of latent syntax (Marginalized) with other recent work: Spitkovsky et al. (2010a) (SAJM'10), Spitkovsky et al. (2010b) (SJA'10), Naseem and Barzilay (2011) (NB'11), and the CS model of Spitkovsky et al. (2013) (SAJ'13).

This suggests that refining the low-resource grammar induction methods may lead to gains in SRL.

Interestingly, the marginalized grammars best the DMV grammar induction method; however, this difference is less pronounced when the DMV is constrained using SRL labels as distant supervision. This could indicate that a better model for grammar induction would result in better performance for SRL. We therefore turn to an analysis of other approaches to grammar induction in Table 8, evaluated on the Penn Treebank. We contrast with methods using distant supervision (Naseem and Barzilay, 2011; Spitkovsky et al., 2010b) and fully unsupervised dependency parsing (Spitkovsky et al., 2013). Following prior work, we exclude punctuation from evaluation and convert the constituency trees to dependencies.[12]

The approach from Spitkovsky et al. (2013) (SAJ'13) outperforms all other approaches, including our marginalized settings. We therefore may be able to achieve further gains in the pipeline model by considering better models of latent syntax, or better search techniques that break out of local optima. Similarly, improving the non-convex optimization of our latent-variable CRF (Marginalized) may offer further gains.

[12]Naseem and Barzilay (2011) and our results use the Penn converter (Pierre and Heiki-Jaan, 2007). Spitkovsky et al. (2010b; 2013) use Collins (1999) head percolation rules.

5 Discussion and Future Work

We have compared various approaches for low-resource semantic role labeling at the state-of-the-art level. We find that we can outperform prior work in the low-resource setting by coupling the selection of feature templates based on information gain with a joint model that marginalizes over latent syntax.

We utilize unlabeled data in both generative and discriminative models for dependency syntax and in generative word clustering. Our discriminative joint models treat latent syntax as a structured-feature to be optimized for the end-task of SRL, while our other grammar induction techniques optimize for unlabeled data likelihood—optionally with distant supervision. We observe that careful use of these unlabeled data resources can improve performance on the end task.

Our subtractive experiments suggest that lemma annotations, a high-resource annotation, may not provide a large benefit for SRL. Our grammar induction analysis indicates that relatively low accuracy can still result in reasonable SRL predictions; still, the models do not outperform those that use supervised syntax, and we aim to explore how well the pipeline models in particular improve when we apply higher accuracy unsupervised grammar induction techniques.

We have utilized well studied datasets in order to best understand the quality of our models relative to prior work. In future work, we hope to explore the effectiveness of our approaches on truly low resource settings by using crowdsourcing to develop semantic role datasets in other languages and domains.

Acknowledgments We thank Richard Johansson, Dennis Mehay, and Stephen Boxwell for help with data. We also thank Jason Naradowsky, Jason Eisner, and anonymous reviewers for comments on the paper.

References

Alfred V. Aho and Jeffrey D. Ullman. 1972. *The Theory of Parsing, Translation, and Compiling*. Prentice-Hall, Inc.

Rami Al-Rfou', Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual NLP. In *Proceedings of the 17th Conference on Computational Natural Language Learning (CoNLL 2013)*. Association for Computational Linguistics.

Anders Björkelund, Love Hafdell, and Pierre Nugues. 2009. Multilingual semantic role labeling. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*. Association for Computational Linguistics.

Stephen Boxwell and Michael White. 2008. Projecting propbank roles onto the CCGbank. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2008)*. European Language Resources Association.

Stephen Boxwell, Chris Brew, Jason Baldridge, Dennis Mehay, and Sujith Ravi. 2011. Semantic role labeling without treebanks? In *Proceedings of the 5th International Joint Conference on Natural Language Processing (IJCNLP)*. Asian Federation of Natural Language Processing.

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

Trevor Cohn, Phil Blunsom, and Sharon Goldwater. 2010. Inducing tree-substitution grammars. *The Journal of Machine Learning Research*, 11.

Michael Collins. 1999. *Head-driven statistical models for natural language parsing*. Ph.D. thesis, University of Pennsylvania.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12.

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 *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*. Association for Computational Linguistics.

Julia Hockenmaier and Mark Steedman. 2007. CCGbank: a corpus of CCG derivations and dependency structures extracted from the Penn Treebank. *Computational Linguistics*, 33(3).

Richard Johansson and Pierre Nugues. 2008. Dependency-based semantic role labeling of PropBank. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*. Association for Computational Linguistics.

Dan Klein and Christopher Manning. 2004. Corpus-Based induction of syntactic structure: Models of dependency and constituency. In *Proceedings of the 42nd Meeting of the Association for Computational Linguistics (ACL 2004)*. Association for Computational Linguistics.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL-08: HLT*. Association for Computational Linguistics.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML 2001)*. Morgan Kaufmann.

Xavier Lluís, Xavier Carreras, and Lluís Màrquez. 2013. Joint arc-factored parsing of syntactic and semantic dependencies. *Transactions of the Association for Computational Linguistics (TACL)*.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The Penn Treebank. *Computational linguistics*, 19(2).

Andre Martins, Noah Smith, Mario Figueiredo, and Pedro Aguiar. 2011. Structured sparsity in structured prediction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*. Association for Computational Linguistics.

R. McDonald, K. Crammer, and F. Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*. Association for Computational Linguistics.

Scott Miller, Jethran Guinness, and Alex Zamanian. 2004. Name tagging with word clusters and discriminative training. In Susan Dumais, Daniel Marcu, and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*. Association for Computational Linguistics.

Jason Naradowsky, Sebastian Riedel, and David Smith. 2012. Improving NLP through marginalization of hidden syntactic structure. In *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing (EMNLP 2012)*. Association for Computational Linguistics.

Tahira Naseem and Regina Barzilay. 2011. Using semantic cues to learn syntax. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI 2011)*. AAAI Press.

Fernando Pereira and Yves Schabes. 1992. Inside-outside reestimation from partially bracketed corpora. In *Proceedings of the 30th Annual Meeting of the Association for Computational Linguistics (ACL 1992)*.

Nugues Pierre and Kalep Heiki-Jaan. 2007. Extended constituent-to-dependency conversion for english. *NODALIDA 2007 Proceedings*.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2).

David A. Smith and Jason Eisner. 2008. Dependency parsing by belief propagation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2008)*. Association for Computational Linguistics.

Valentin I. Spitkovsky, Hiyan Alshawi, Daniel Jurafsky, and Christopher D Manning. 2010a. Viterbi training improves unsupervised dependency parsing. In *Proceedings of the 14th Conference on Computational Natural Language Learning (CoNLL 2010)*. Association for Computational Linguistics.

Valentin I. Spitkovsky, Daniel Jurafsky, and Hiyan Alshawi. 2010b. Profiting from mark-up: Hyper-text annotations for guided parsing. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*. Association for Computational Linguistics.

Valentin I. Spitkovsky, Hiyan Alshawi, Angel X. Chang, and Daniel Jurafsky. 2011. Unsupervised dependency parsing without gold part-of-speech tags. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing (EMNLP 2011)*. Association for Computational Linguistics.

Valentin I. Spitkovsky, Hiyan Alshawi, and Daniel Jurafsky. 2013. Breaking out of local optima with count transforms and model recombination: A study in grammar induction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (EMNLP 2013)*. Association for Computational Linguistics.

Kristina Toutanova, Aria Haghighi, and Christopher Manning. 2005. Joint learning improves semantic role labeling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics (ACL 2005)*. Association for Computational Linguistics.

Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. 2009. Feature hashing for large scale multitask learning. In Léon Bottou and Michael Littman, editors, *Proceedings of the 26th Annual International Conference on Machine Learning (ICML 2009)*. Omnipress.

Daniel H. Younger. 1967. Recognition and parsing of context-free languages in time n^3. *Information and Control*, 10(2).

Hai Zhao, Wenliang Chen, Chunyu Kity, and Guodong Zhou. 2009. Multilingual dependency learning: A huge feature engineering method to semantic dependency parsing. In *Proceedings of the 13th Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*. Association for Computational Linguistics.