A Bayesian Approach to Unsupervised Semantic Role Induction

Ivan Titov **Alexandre Klementiev**
Saarland University
Saarbrücken, Germany
{titov|aklement}@mmci.uni-saarland.de

Abstract

We introduce two Bayesian models for unsupervised semantic role labeling (SRL) task. The models treat SRL as clustering of syntactic signatures of arguments with clusters corresponding to semantic roles. The first model induces these clusterings independently for each predicate, exploiting the Chinese Restaurant Process (CRP) as a prior. In a more refined hierarchical model, we inject the intuition that the clusterings are similar across different predicates, even though they are not necessarily identical. This intuition is encoded as a distance-dependent CRP with a distance between two syntactic signatures indicating how likely they are to correspond to a single semantic role. These distances are automatically induced within the model and shared across predicates. Both models achieve state-of-the-art results when evaluated on PropBank, with the coupled model consistently outperforming the factored counterpart in all experimental set-ups.

1 Introduction

Semantic role labeling (SRL) (Gildea and Jurafsky, 2002), a shallow semantic parsing task, has recently attracted a lot of attention in the computational linguistic community (Carreras and Màrquez, 2005; Surdeanu et al., 2008; Hajič et al., 2009). The task involves prediction of predicate argument structure, i.e. both identification of arguments as well as assignment of labels according to their underlying *semantic role*. For example, in the following sentences:

(a) [$_{A0}$ Mary] opened [$_{A1}$ the door].

(b) [$_{A0}$ Mary] is expected to open [$_{A1}$ the door].

(c) [$_{A1}$ The door] opened.

(d) [$_{A1}$ The door] was opened [$_{A0}$ by Mary].

Mary always takes an agent role ($A0$) for the predicate *open*, and *door* is always a patient ($A1$). SRL representations have many potential applications in natural language processing and have recently been shown to be beneficial in question answering (Shen and Lapata, 2007; Kaisser and Webber, 2007), textual entailment (Sammons et al., 2009), machine translation (Wu and Fung, 2009; Liu and Gildea, 2010; Wu et al., 2011; Gao and Vogel, 2011), and dialogue systems (Basili et al., 2009; van der Plas et al., 2011), among others. Though syntactic representations are often predictive of semantic roles (Levin, 1993), the interface between syntactic and semantic representations is far from trivial. The lack of simple deterministic rules for mapping syntax to shallow semantics motivates the use of statistical methods.

Although current statistical approaches have been successful in predicting shallow semantic representations, they typically require large amounts of annotated data to estimate model parameters. These resources are scarce and expensive to create, and even the largest of them have low coverage (Palmer and Sporleder, 2010). Moreover, these models are domain-specific, and their performance drops substantially when they are used in a new domain (Pradhan et al., 2008). Such domain specificity is arguably unavoidable for a semantic analyzer, as even the definitions of semantic roles are typically predicate specific, and different domains can have radically different distributions of predicates (and their senses). The necessity for a large amounts of human-annotated data for every language and domain is one of the major obstacles to the wide-spread adoption of semantic role representations.

These challenges motivate the need for unsupervised methods which, instead of relying on labeled data, can exploit large amounts of unlabeled texts. In this paper, we propose simple and effi-

cient hierarchical Bayesian models for this task.

It is natural to split the SRL task into two stages: the identification of arguments (the *identification* stage) and the assignment of semantic roles (the *labeling* stage). In this and in much of the previous work on unsupervised techniques, the focus is on the labeling stage. Identification, though an important problem, can be tackled with heuristics (Lang and Lapata, 2011a; Grenager and Manning, 2006) or, potentially, by using a supervised classifier trained on a small amount of data. We follow (Lang and Lapata, 2011a), and regard the labeling stage as clustering of syntactic signatures of argument realizations for every predicate. In our first model, as in most of the previous work on unsupervised SRL, we define an independent model for each predicate. We use the Chinese Restaurant Process (CRP) (Ferguson, 1973) as a prior for the clustering of syntactic signatures. The resulting model achieves state-of-the-art results, substantially outperforming previous methods evaluated in the same setting.

In the first model, for each predicate we independently induce a linking between syntax and semantics, encoded as a clustering of syntactic signatures. The clustering implicitly defines the set of permissible *alternations*, or changes in the syntactic realization of the argument structure of the verb. Though different verbs admit different alternations, some alternations are shared across multiple verbs and are very frequent (e.g., passivization, example sentences (a) vs. (d), or dativization: *John gave a book to Mary* vs. *John gave Mary a book*) (Levin, 1993). Therefore, it is natural to assume that the clusterings should be similar, though not identical, across verbs.

Our second model encodes this intuition by replacing the CRP prior for each predicate with a distance-dependent CRP (dd-CRP) prior (Blei and Frazier, 2011) shared across predicates. The distance between two syntactic signatures encodes how likely they are to correspond to a single semantic role. Unlike most of the previous work exploiting distance-dependent CRPs (Blei and Frazier, 2011; Socher et al., 2011; Duan et al., 2007), we do not encode prior or external knowledge in the distance function but rather induce it automatically within our Bayesian model. The coupled dd-CRP model consistently outperforms the factored CRP counterpart across all the experimental settings (with gold and predicted syntactic parses, and with gold and automatically identified arguments).

Both models admit efficient inference: the estimation time on the Penn Treebank WSJ corpus does not exceed 30 minutes on a single processor and the inference algorithm is highly parallelizable, reducing inference time down to several minutes on multiple processors. This suggests that the models scale to much larger corpora, which is an important property for a successful unsupervised learning method, as unlabeled data is abundant.

The rest of the paper is structured as follows. Section 2 begins with a definition of the semantic role labeling task and discuss some specifics of the unsupervised setting. In Section 3, we describe CRPs and dd-CRPs, the key components of our models. In Sections 4 – 6, we describe our factored and coupled models and the inference method. Section 7 provides both evaluation and analysis. Finally, additional related work is presented in Section 8.

2 Task Definition

In this work, instead of assuming the availability of role annotated data, we rely only on automatically generated syntactic dependency graphs. While we cannot expect that syntactic structure can trivially map to a semantic representation (Palmer et al., 2005)[1], we can use syntactic cues to help us in both stages of unsupervised SRL. Before defining our task, let us consider the two stages separately.

In the argument identification stage, we implement a heuristic proposed in (Lang and Lapata, 2011a) comprised of a list of 8 rules, which use nonlexicalized properties of syntactic paths between a predicate and a candidate argument to iteratively discard non-arguments from the list of all words in a sentence. Note that inducing these rules for a new language would require some linguistic expertise. One alternative may be to annotate a small number of arguments and train a classifier with nonlexicalized features instead.

In the argument labeling stage, semantic roles are represented by clusters of arguments, and labeling a particular argument corresponds to deciding on its role cluster. However, instead of deal-

[1] Although it provides a strong baseline which is difficult to beat (Grenager and Manning, 2006; Lang and Lapata, 2010; Lang and Lapata, 2011a).

ing with argument occurrences directly, we represent them as predicate specific syntactic signatures, and refer to them as *argument keys*. This representation aids our models in inducing high purity clusters (of argument keys) while reducing their granularity. We follow (Lang and Lapata, 2011a) and use the following syntactic features to form the argument key representation:

- Active or passive verb voice (`ACT/PASS`).
- Argument position relative to predicate (`LEFT/RIGHT`).
- Syntactic relation to its governor.
- Preposition used for argument realization.

In the example sentences in Section 1, the argument keys for candidate arguments *Mary* for sentences (a) and (d) would be `ACT:LEFT:SBJ` and `PASS:RIGHT:LGS->by`,[2] respectively. While aiming to increase the purity of argument key clusters, this particular representation will not always produce a good match: e.g. *the door* in sentence (c) will have the same key as *Mary* in sentence (a). Increasing the expressiveness of the argument key representation by flagging intransitive constructions would distinguish that pair of arguments. However, we keep this particular representation, in part to compare with the previous work.

In this work, we treat the unsupervised semantic role labeling task as *clustering of argument keys*. Thus, argument occurrences in the corpus whose keys are clustered together are assigned the same semantic role. Note that some adjunct-like modifier arguments are already explicitly represented in syntax and thus do not need to be clustered (modifiers `AM-TMP`, `AM-MNR`, `AM-LOC`, and `AM-DIR` are encoded as 'syntactic' relations `TMP`, `MNR`, `LOC`, and `DIR`, respectively (Surdeanu et al., 2008)); instead we directly use the syntactic labels as semantic roles.

3 Traditional and Distance-dependent CRPs

The central components of our non-parametric Bayesian models are the Chinese Restaurant Processes (CRPs) and the closely related Dirichlet Processes (DPs) (Ferguson, 1973).

CRPs define probability distributions over partitions of a set of objects. An intuitive metaphor for describing CRPs is assignment of tables to restaurant customers. Assume a restaurant with a sequence of tables, and customers who walk into the restaurant one at a time and choose a table to join. The first customer to enter is assigned the first table. Suppose that when a client number i enters the restaurant, $i-1$ customers are sitting at each of the $k \in (1, \ldots, K)$ tables occupied so far. The new customer is then either seated at one of the K tables with probability $\frac{N_k}{i-1+\alpha}$, where N_k is the number customers already sitting at table k, or assigned to a new table with the probability $\frac{\alpha}{i-1+\alpha}$. The concentration parameter α encodes the granularity of the drawn partitions: the larger α, the larger the expected number of occupied tables. Though it is convenient to describe CRP in a sequential manner, the probability of a seating arrangement is invariant of the order of customers' arrival, i.e. the process is *exchangeable*. In our factored model, we use CRPs as a prior for clustering argument keys, as we explain in Section 4.

Often CRP is used as a part of the Dirichlet Process mixture model where each subset in the partition (each table) selects a parameter (a meal) from some base distribution over parameters. This parameter is then used to generate all data points corresponding to customers assigned to the table. The Dirichlet processes (DP) are closely connected to CRPs: instead of choosing meals for customers through the described generative story, one can equivalently draw a distribution G over meals from DP and then draw a meal for every customer from G. We refer the reader to Teh (2010) for details on CRPs and DPs. In our method, we use DPs to model distributions of arguments for every role.

In order to clarify how similarities between customers can be integrated in the generative process, we start by reformulating the traditional CRP in an equivalent form so that distance-dependent CRP (dd-CRP) can be seen as its generalization. Instead of selecting a table for each customer as described above, one can equivalently assume that a customer i chooses one of the previous customers c_i as a partner with probability $\frac{1}{i-1+\alpha}$ and sits at the same table, or occupies a new table with the probability $\frac{\alpha}{i-1+\alpha}$. The transitive closure of this seating-with relation determines the partition.

A generalization of this view leads to the definition of the distance-dependent CRP. In dd-CRPs,

[2] `LGS` denotes a logical subject in a passive construction (Surdeanu et al., 2008).

a customer i chooses a partner $c_i = j$ with the probability proportional to some non-negative score $d_{i,j}$ ($d_{i,j} = d_{j,i}$) which encodes a similarity between the two customers.[3] More formally,

$$p(c_i = j|D, \alpha) \propto \begin{cases} d_{i,j}, & i \neq j \\ \alpha, & i = j \end{cases} \quad (1)$$

where D is the entire similarity graph. This process lacks the exchangeability property of the traditional CRP but efficient approximate inference with dd-CRP is possible with Gibbs sampling. For more details on inference with dd-CRPs, we refer the reader to Blei and Frazier (2011).

Though in previous work dd-CRP was used either to encode prior knowledge (Blei and Frazier, 2011) or other external information (Socher et al., 2011), we treat D as a latent variable drawn from some prior distribution over weighted graphs. This view provides a powerful approach for coupling a family of distinct but similar clusterings: the family of clusterings can be drawn by first choosing a similarity graph D for the entire family and then re-using D to generate each of the clusterings independently of each other as defined by equation (1). In Section 5, we explain how we use this formalism to encode relatedness between argument key clusterings for different predicates.

4 Factored Model

In this section we describe the *factored* method which models each predicate independently. In Section 2 we defined our task as clustering of argument keys, where each cluster corresponds to a semantic role. If an argument key k is assigned to a role r ($k \in r$), all of its occurrences are labeled r.

Our Bayesian model encodes two common assumptions about semantic roles. First, we enforce the selectional restriction assumption: we assume that the distribution over potential argument fillers is sparse for every role, implying that 'peaky' distributions of arguments for each role r are preferred to flat distributions. Second, each role normally appears at most once per predicate occurrence. Our inference will search for a clustering which meets the above requirements to the maximal extent.

[3]It may be more standard to use a decay function $f : \mathcal{R} \rightarrow \mathcal{R}$ and choose a partner with the probability proportional to $f(-d_{i,j})$. However, the two forms are equivalent and using scores $d_{i,j}$ directly is more convenient for our induction purposes.

Our model associates two distributions with each predicate: one governs the selection of argument fillers for each semantic role, and the other models (and penalizes) duplicate occurrence of roles. Each predicate occurrence is generated independently given these distributions. Let us describe the model by first defining how the set of model parameters and an argument key clustering are drawn, and then explaining the generation of individual predicate and argument instances. The generative story is formally presented in Figure 1.

We start by generating a partition of argument keys B_p with each subset $r \in B_p$ representing a single semantic role. The partitions are drawn from CRP(α) (see the **Factored model** section of Figure 1) independently for each predicate. The crucial part of the model is the set of selectional preference parameters $\theta_{p,r}$, the distributions of arguments x for each role r of predicate p. We represent arguments by their syntactic heads,[4] or more specifically, by either their lemmas or word clusters assigned to the head by an external clustering algorithm, as we will discuss in more detail in Section 7.[5] For the agent role $A0$ of the predicate *open*, for example, this distribution would assign most of the probability mass to arguments denoting sentient beings, whereas the distribution for the patient role $A1$ would concentrate on arguments representing "openable" things (doors, boxes, books, etc).

In order to encode the assumption about sparseness of the distributions $\theta_{p,r}$, we draw them from the DP prior $DP(\beta, H^{(A)})$ with a small concentration parameter β, the base probability distribution $H^{(A)}$ is just the normalized frequencies of arguments in the corpus. The geometric distribution $\psi_{p,r}$ is used to model the number of times a role r appears with a given predicate occurrence. The decision whether to generate at least one role r is drawn from the uniform Bernoulli distribution. If 0 is drawn then the semantic role is not realized for the given occurrence, otherwise the number of additional roles r is drawn from the geometric distribution $Geom(\psi_{p,r})$. The Beta priors over ψ

[4]For prepositional phrases, we take as head the head noun of the object noun phrase as it encodes crucial lexical information. However, the preposition is not ignored but rather encoded in the corresponding argument key, as explained in Section 2.

[5]Alternatively, the clustering of arguments could be induced within the model, as done in (Titov and Klementiev, 2011).

Clustering of argument keys:

Factored model:
for each predicate $p = 1, 2, \ldots$:
$B_p \sim CRP(\alpha)$ [partition of arg keys]

Coupled model:
$D \sim NonInform$ [similarity graph]
for each predicate $p = 1, 2, \ldots$:
$B_p \sim dd\text{-}CRP(\alpha, D)$ [partition of arg keys]

Parameters:

for each predicate $p = 1, 2, \ldots$:
for each role $r \in B_p$:
$\theta_{p,r} \sim DP(\beta, H^{(A)})$ [distrib of arg fillers]
$\psi_{p,r} \sim Beta(\eta_0, \eta_1)$ [geom distr for dup roles]

Data Generation:

for each predicate $p = 1, 2, \ldots$:
for each occurrence l of p:
for every role $r \in B_p$:
if $[n \sim Unif(0, 1)] = 1$: [role appears at least once]
GenArgument(p, r) [draw one arg]
while $[n \sim \psi_{p,r}] = 1$: [continue generation]
GenArgument(p, r) [draw more args]

GenArgument(p, r):
$k_{p,r} \sim Unif(1, \ldots, |r|)$ [draw arg key]
$x_{p,r} \sim \theta_{p,r}$ [draw arg filler]

Figure 1: Generative stories for the factored and coupled models.

can indicate the preference towards generating at most one argument for each role. For example, it would express the preference that a predicate *open* typically appears with a single agent and a single patient arguments.

Now, when parameters and argument key clusterings are chosen, we can summarize the remainder of the generative story as follows. We begin by independently drawing occurrences for each predicate. For each predicate role we independently decide on the number of role occurrences. Then we generate each of the arguments (see **GenArgument**) by generating an argument key $k_{p,r}$ uniformly from the set of argument keys assigned to the cluster r, and finally choosing its filler $x_{p,r}$, where the filler is either a lemma or a word cluster corresponding to the syntactic head of the argument.

5 Coupled Model

As we argued in Section 1, clusterings of argument keys implicitly encode the pattern of alternations for a predicate. E.g., passivization can be roughly represented with the clustering of the key `ACT:LEFT:SBJ` with `PASS:RIGHT:LGS->by` and `ACT:RIGHT:OBJ` with `PASS:LEFT:SBJ`. The set of permissible alternations is predicate-specific,[6] but nevertheless they arguably represent a small subset of all clusterings of argument keys. Also, some alternations are more likely to be applicable to a verb than others: for example, passivization and dativization alternations are both fairly frequent, whereas, locative-preposition-drop alternation (*Mary climbed up the mountain* vs. *Mary climbed the mountain*) is less common and applicable only to several classes of predicates representing motion (Levin, 1993). We represent this observation by quantifying how likely a pair of keys is to be clustered. These scores ($d_{i,j}$ for every pair of argument keys i and j) are induced automatically within the model, and treated as latent variables shared across predicates. Intuitively, if data for several predicates strongly suggests that two argument keys should be clustered (e.g., there is a large overlap between argument fillers for the two keys) then the posterior will indicate that $d_{i,j}$ is expected to be greater for the pair $\{i, j\}$ than for some other pair $\{i', j'\}$ for which the evidence is less clear. Consequently, argument keys i and j will be clustered even for predicates without strong evidence for such a clustering, whereas i' and j' will not.

One argument against coupling predicates may stem from the fact that we are using unlabeled data and may be able to obtain sufficient amount of learning material even for less frequent predicates. This may be a valid observation, but another rationale for sharing this similarity structure is the hypothesis that alternations may be easier to detect for some predicates than for others. For example, argument key clustering of predicates with very restrictive selectional restrictions on argument fillers is presumably easier than clustering for predicates with less restrictive and overlapping selectional restriction, as compactness of selectional preferences is a central assumption driving unsupervised learning of semantic roles. E.g., predicates *change* and *defrost* belong to the same Levin class (*change-of-state verbs*) and therefore admit similar alternations. However, the set of potential patients of *defrost* is sufficiently restricted,

[6] Or, at least specific to a class of predicates (Levin, 1993).

whereas the selectional restrictions for the patient of *change* are far less specific and they overlap with selectional restrictions for the agent role, further complicating the clustering induction task. This observation suggests that sharing clustering preferences across verbs is likely to help even if the unlabeled data is plentiful for every predicate.

More formally, we generate scores $d_{i,j}$, or equivalently, the full labeled graph D with vertices corresponding to argument keys and edges weighted with the similarity scores, from a prior. In our experiments we use a non-informative prior which factorizes over pairs (i.e. edges of the graph D), though more powerful alternatives can be considered. Then we use it, in a dd-CRP(α, D), to generate clusterings of argument keys for every predicate. The rest of the generative story is the same as for the factored model. The part relevant to this model is shown in the **Coupled model** section of Figure 1.

Note that this approach does not assume that the frequencies of syntactic patterns corresponding to alternations are similar, and a large value for $d_{i,j}$ does not necessarily mean that the corresponding syntactic frames i and j are very frequent in a corpus. What it indicates is that a large number of different predicates undergo the corresponding alternation; the frequency of the alternation is a different matter. We believe that this is an important point, as we do not make a restricting assumption that an alternation has the same distributional properties for all verbs which undergo this alternation.

6 Inference

An inference algorithm for an unsupervised model should be efficient enough to handle vast amounts of unlabeled data, as it can easily be obtained and is likely to improve results. We use a simple approximate inference algorithm based on greedy MAP search. We start by discussing MAP search for argument key clustering with the factored model and then discuss its extension applicable to the coupled model.

6.1 Role Induction

For the factored model, semantic roles for every predicate are induced independently. Nevertheless, search for a MAP clustering can be expensive, as even a move involving a single argument key implies some computations for all its occurrences in the corpus. Instead of more complex MAP search algorithms (see, e.g., (Daume III, 2007)), we use a greedy procedure where we start with each argument key assigned to an individual cluster, and then iteratively try to merge clusters. Each move involves (1) choosing an argument key and (2) deciding on a cluster to reassign it to. This is done by considering all clusters (including creating a new one) and choosing the most probable one.

Instead of choosing argument keys randomly at the first stage, we order them by corpus frequency. This ordering is beneficial as getting clustering right for frequent argument keys is more important and the corresponding decisions should be made earlier.[7] We used a single iteration in our experiments, as we have not noticed any benefit from using multiple iterations.

6.2 Similarity Graph Induction

In the coupled model, clusterings for different predicates are statistically dependent, as the similarity structure D is latent and shared across predicates. Consequently, a more complex inference procedure is needed. For simplicity here and in our experiments, we use the non-informative prior distribution over D which assigns the same prior probability to every possible weight $d_{i,j}$ for every pair $\{i, j\}$.

Recall that the dd-CRP prior is defined in terms of customers choosing other customers to sit with. For the moment, let us assume that this relation among argument keys is known, that is, every argument key k for predicate p has chosen an argument key $c_{p,k}$ to 'sit' with. We can compute the MAP estimate for all $d_{i,j}$ by maximizing the objective:

$$\arg\max_{d_{i,j},\, i \neq j} \sum_{p} \sum_{k \in \boldsymbol{K}_p} \log \frac{d_{k,c_{p,k}}}{\sum_{k' \in \boldsymbol{K}_p} d_{k,k'}},$$

where $\boldsymbol{K}_p$ is the set of all argument keys for the predicate p. We slightly abuse the notation by using $d_{i,i}$ to denote the concentration parameter α in the previous expression. Note that we also assume that similarities are symmetric, $d_{i,j} = d_{j,i}$. If the set of argument keys $\boldsymbol{K}_p$ would be the same for every predicate, then the optimal $d_{i,j}$ would

[7] This idea has been explored before for shallow semantic representations (Lang and Lapata, 2011a; Titov and Klementiev, 2011).

be proportional to the number of times either i selects j as a partner, or j chooses i as a partner.[8] This no longer holds if the sets are different, but the solution can be found efficiently using a numeric optimization strategy; we use the gradient descent algorithm.

We do not learn the concentration parameter α, as it is used in our model to indicate the desired granularity of semantic roles, but instead only learn $d_{i,j}$ ($i \neq j$). However, just learning the concentration parameter would not be sufficient as the effective concentration can be reduced or increased arbitrarily by scaling all the similarities $d_{i,j}$ ($i \neq j$) at once, as follows from expression (1). Instead, we enforce the normalization constraint on the similarities $d_{i,j}$. We ensure that the prior probability of choosing itself as a partner, averaged over predicates, is the same as it would be with uniform $d_{i,j}$ ($d_{i,j} = 1$ for every key pair $\{i, j\}$, $i \neq j$). This roughly says that we want to preserve the same granularity of clustering as it was with the uniform similarities. We accomplish this normalization in a post-hoc fashion by dividing the weights after optimization by $\sum_p \sum_{k,k' \in \boldsymbol{K}_p,\, k' \neq k} d_{k,k'} / \sum_p |\boldsymbol{K}_p|(|\boldsymbol{K}_p| - 1)$.

If D is fixed, partners for every predicate p and every k can be found using virtually the same algorithm as in Section 6.1: the only difference is that, instead of a cluster, each argument key iteratively chooses a partner.

Though, in practice, both the choice of partners and the similarity graphs are latent, we can use an iterative approach to obtain a joint MAP estimate of c_k (for every k) and the similarity graph D by alternating the two steps.[9]

Notice that the resulting algorithm is again highly parallelizable: the graph induction stage is fast, and induction of the seat-with relation (i.e. clustering argument keys) is factorizable over predicates.

One shortcoming of this approach is typical for generative models with multiple 'features': when such a model predicts a latent variable, it tends to ignore the prior class distribution and relies solely on features. This behavior is due to the over-simplifying independence assumptions. It is well known, for instance, that the posterior with Naive Bayes tends to be overconfident due to violated conditional independence assumptions (Rennie, 2001). The same behavior is observed here: the shared prior does not have sufficient effect on frequent predicates.[10] Though different techniques have been developed to discount the over-confidence (Kolcz and Chowdhury, 2005), we use the most basic one: we raise the likelihood term in power $\frac{1}{T}$, where the parameter T is chosen empirically.

7 Empirical Evaluation

7.1 Data and Evaluation

We keep the general setup of (Lang and Lapata, 2011a), to evaluate our models and compare them to the current state of the art. We run all of our experiments on the standard CoNLL 2008 shared task (Surdeanu et al., 2008) version of Penn Treebank WSJ and PropBank. In addition to gold dependency analyses and gold PropBank annotations, it has dependency structures generated automatically by the MaltParser (Nivre et al., 2007). We vary our experimental setup as follows:

- We evaluate our models on gold and automatically generated parses, and use either gold PropBank annotations or the heuristic from Section 2 to identify arguments, resulting in four experimental regimes.
- In order to reduce the sparsity of predicate argument fillers we consider replacing lemmas of their syntactic heads with word clusters induced by a clustering algorithm as a preprocessing step. In particular, we use Brown (*Br*) clustering (Brown et al., 1992) induced over RCV1 corpus (Turian et al., 2010). Although the clustering is hierarchical, we only use a cluster at the lowest level of the hierarchy for each word.

We use the purity (PU) and collocation (CO) metrics as well as their harmonic mean (F1) to measure the quality of the resulting clusters. Purity measures the degree to which each cluster contains arguments sharing the same gold role:

$$PU = \frac{1}{N} \sum_i \max_j |G_j \cap C_i|$$

where if C_i is the set of arguments in the i-th induced cluster, G_j is the set of arguments in the jth

[8] Note that weights $d_{i,j}$ are invariant under rescaling when the rescaling is also applied to the concentration parameter α.

[9] In practice, two iterations were sufficient.

[10] The coupled model without discounting still outperforms the factored counterpart in our experiments.

gold cluster, and N is the total number of arguments. Collocation evaluates the degree to which arguments with the same gold roles are assigned to a single cluster. It is computed as follows:

$$CO = \frac{1}{N} \sum_j \max_i |G_j \cap C_i|$$

We compute the aggregate PU, CO, and F1 scores over all predicates in the same way as (Lang and Lapata, 2011a) by weighting the scores of each predicate by the number of its argument occurrences. Note that since our goal is to evaluate the clustering algorithms, we *do not* include incorrectly identified arguments (i.e. mistakes made by the heuristic defined in Section 2) when computing these metrics.

We evaluate both factored and coupled models proposed in this work with and without Brown word clustering of argument fillers (*Factored*, *Coupled*, *Factored+Br*, *Coupled+Br*). Our models are robust to parameter settings, they were tuned (to an order of magnitude) on the development set and were the same for all model variants: $\alpha = 1.e\text{-}3$, $\beta = 1.e\text{-}3$, $\eta_0 = 1.e\text{-}3$, $\eta_1 = 1.e\text{-}10$, $T = 5$. Although they can be induced within the model, we set them by hand to indicate granularity preferences. We compare our results with the following alternative approaches. The syntactic function baseline (*SyntF*) simply clusters predicate arguments according to the dependency relation to their head. Following (Lang and Lapata, 2010), we allocate a cluster for each of 20 most frequent relations in the CoNLL dataset and one cluster for all other relations. We also compare our performance with the Latent Logistic classification (Lang and Lapata, 2010), Split-Merge clustering (Lang and Lapata, 2011a), and Graph Partitioning (Lang and Lapata, 2011b) approaches (labeled *LLogistic*, *SplitMerge*, and *GraphPart*, respectively) which achieve the current best unsupervised SRL results in this setting.

7.2 Results

7.2.1 Gold Arguments

Experimental results are summarized in Table 1. We begin by comparing our models to the three existing clustering approaches on gold syntactic parses, and using gold PropBank annotations to identify predicate arguments. In this set of experiments we measure the relative performance of argument clustering, removing the identification stage, and minimize the noise due to automatic syntactic annotations. All four variants of the models we propose substantially outperform other models: the coupled model with Brown clustering of argument fillers (*Coupled+Br*) beats the previous best model *SplitMerge* by 2.9% F1 score. As mentioned in Section 2, our approach specifically does not cluster some of the modifier arguments. In order to verify that this and argument filler clustering were not the only aspects of our approach contributing to performance improvements, we also evaluated our coupled model without Brown clustering and treating modifiers as regular arguments. The model achieves 89.2% purity, 74.0% collocation, and 80.9% F1 scores, still substantially outperforming all of the alternative approaches. Replacing gold parses with MaltParser analyses we see a similar trend, where *Coupled+Br* outperforms the best alternative approach *SplitMerge* by 1.5%.

	gold parses			auto parses		
	PU	CO	F1	PU	CO	F1
LLogistic	79.5	76.5	78.0	77.9	74.4	76.2
SplitMerge	88.7	73.0	80.1	86.5	69.8	77.3
GraphPart	88.6	70.7	78.6	87.4	65.9	75.2
Factored	88.1	77.1	82.2	85.1	71.8	77.9
Coupled	89.3	76.6	82.5	86.7	71.2	78.2
Factored+Br	86.8	78.8	82.6	83.8	74.1	78.6
Coupled+Br	88.7	78.1	**83.0**	86.2	72.7	**78.8**
SyntF	81.6	77.5	79.5	77.1	70.9	73.9

Table 1: Argument clustering performance with *gold argument identification*. Bold-face is used to highlight the best F1 scores.

7.2.2 Automatic Arguments

Results are summarized in Table 2.[11] The precision and recall of our re-implementation of the argument identification heuristic described in Section 2 on gold parses were 87.7% and 88.0%, respectively, and do not quite match 88.1% and 87.9% reported in (Lang and Lapata, 2011a). Since we could not reproduce their argument identification stage exactly, we are omitting their results for the two regimes, instead including the results for our two best models *Factored+Br* and *Coupled+Br*. We see a similar trend, where the coupled system consistently outperforms its factored counterpart, achieving 85.8% and 83.9% F1

[11] Note, that the scores are computed on correctly identified arguments only, and tend to be higher in these experiments probably because the complex arguments get discarded by the heuristic.

	gold parses			auto parses		
	PU	CO	F1	PU	CO	F1
Factored+Br	87.8	82.9	85.3	85.8	81.1	83.4
Coupled+Br	89.2	82.6	**85.8**	87.4	80.7	**83.9**
SyntF	83.5	81.4	82.4	81.4	79.1	80.2

Table 2: Argument clustering performance with *automatic argument identification*.

for gold and MaltParser analyses, respectively.

We observe that consistently through the four regimes, sharing of alternations between predicates captured by the coupled model outperforms the factored version, and that reducing the argument filler sparsity with clustering also has a substantial positive effect. Due to the space constraints we are not able to present detailed analysis of the induced similarity graph D, however, argument-key pairs with the highest induced similarity encode, among other things, passivization, benefactive alternations, near-interchangeability of some subordinating conjunctions and prepositions (e.g., *if* and *whether*), as well as, restoring some of the unnecessary splits introduced by the argument key definition (e.g., semantic roles for adverbials do not normally depend on whether the construction is passive or active).

8 Related Work

Most of SRL research has focused on the supervised setting (Carreras and Màrquez, 2005; Surdeanu et al., 2008), however, lack of annotated resources for most languages and insufficient coverage provided by the existing resources motivates the need for using unlabeled data or other forms of weak supervision. This work includes methods based on graph alignment between labeled and unlabeled data (Fürstenau and Lapata, 2009), using unlabeled data to improve lexical generalization (Deschacht and Moens, 2009), and projection of annotation across languages (Pado and Lapata, 2009; van der Plas et al., 2011). Semi-supervised and weakly-supervised techniques have also been explored for other types of semantic representations but these studies have mostly focused on restricted domains (Kate and Mooney, 2007; Liang et al., 2009; Titov and Kozhevnikov, 2010; Goldwasser et al., 2011; Liang et al., 2011).

Unsupervised learning has been one of the central paradigms for the closely-related area of relation extraction, where several techniques have been proposed to cluster semantically similar verbalizations of relations (Lin and Pantel, 2001; Banko et al., 2007). Early unsupervised approaches to the SRL problem include the work by Swier and Stevenson (2004), where the VerbNet verb lexicon was used to guide unsupervised learning, and a generative model of Grenager and Manning (2006) which exploits linguistic priors on syntactic-semantic interface.

More recently, the role induction problem has been studied in Lang and Lapata (2010) where it has been reformulated as a problem of detecting alterations and mapping non-standard linkings to the canonical ones. Later, Lang and Lapata (2011a) proposed an algorithmic approach to clustering argument signatures which achieves higher accuracy and outperforms the syntactic baseline. In Lang and Lapata (2011b), the role induction problem is formulated as a graph partitioning problem: each vertex in the graph corresponds to a predicate occurrence and edges represent lexical and syntactic similarities between the occurrences. Unsupervised induction of semantics has also been studied in Poon and Domingos (2009) and Titov and Klementiev (2010) but the induced representations are not entirely compatible with the PropBank-style annotations and they have been evaluated only on a question answering task for the biomedical domain. Also, the related task of unsupervised argument identification was considered in Abend et al. (2009).

9 Conclusions

In this work we introduced two Bayesian models for unsupervised role induction. They treat the task as a family of related clustering problems, one for each predicate. The first factored model induces each clustering independently, whereas the second model couples them by exploiting a novel technique for sharing clustering preferences across a family of clusterings. Both methods achieve state-of-the-art results with the coupled model outperforming the factored counterpart in all regimes.

Acknowledgements

The authors acknowledge the support of the MMCI Cluster of Excellence, and thank Hagen Fürstenau, Mikhail Kozhevnikov, Alexis Palmer, Manfred Pinkal, Caroline Sporleder and the anonymous reviewers for their suggestions, and Joel Lang for answering questions about their methods and data.

References

Omri Abend, Roi Reichart, and Ari Rappoport. 2009. Unsupervised argument identification for semantic role labeling. In *ACL-IJCNLP*.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *IJCAI*.

Roberto Basili, Diego De Cao, Danilo Croce, Bonaventura Coppola, and Alessandro Moschitti. 2009. Cross-language frame semantics transfer in bilingual corpora. In *CICLING*.

David M. Blei and Peter Frazier. 2011. Distance dependent chinese restaurant processes. *Journal of Machine Learning Research*, 12:2461–2488.

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

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In *CoNLL*.

Hal Daume III. 2007. Fast search for dirichlet process mixture models. In *AISTATS*.

Koen Deschacht and Marie-Francine Moens. 2009. Semi-supervised semantic role labeling using the Latent Words Language Model. In *EMNLP*.

Jason Duan, Michele Guindani, and Alan Gelfand. 2007. Generalized spatial dirichlet process models. *Biometrika*, 94:809–825.

Thomas S. Ferguson. 1973. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230.

Hagen Fürstenau and Mirella Lapata. 2009. Graph alignment for semi-supervised semantic role labeling. In *EMNLP*.

Qin Gao and Stephan Vogel. 2011. Corpus expansion for statistical machine translation with semantic role label substitution rules. In *ACL:HLT*.

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

Dan Goldwasser, Roi Reichart, James Clarke, and Dan Roth. 2011. Confidence driven unsupervised semantic parsing. In *ACL*.

Trond Grenager and Christoph Manning. 2006. Unsupervised discovery of a statistical verb lexicon. In *EMNLP*.

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), June 4-5*.

Michael Kaisser and Bonnie Webber. 2007. Question answering based on semantic roles. In *ACL Workshop on Deep Linguistic Processing*.

Rohit J. Kate and Raymond J. Mooney. 2007. Learning language semantics from ambigous supervision. In *AAAI*.

Aleksander Kolcz and Abdur Chowdhury. 2005. Discounting over-confidence of naive bayes in high-recall text classification. In *ECML*.

Joel Lang and Mirella Lapata. 2010. Unsupervised induction of semantic roles. In *ACL*.

Joel Lang and Mirella Lapata. 2011a. Unsupervised semantic role induction via split-merge clustering. In *ACL*.

Joel Lang and Mirella Lapata. 2011b. Unsupervised semantic role induction with graph partitioning. In *EMNLP*.

Beth Levin. 1993. *English Verb Classes and Alternations: A Preliminary Investigation*. University of Chicago Press.

Percy Liang, Michael I. Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *ACL-IJCNLP*.

Percy Liang, Michael Jordan, and Dan Klein. 2011. Learning dependency-based compositional semantics. In *ACL: HLT*.

Dekang Lin and Patrick Pantel. 2001. DIRT – discovery of inference rules from text. In *KDD*.

Ding Liu and Daniel Gildea. 2010. Semantic role features for machine translation. In *Coling*.

J. Nivre, J. Hall, S. Kübler, R. McDonald, J. Nilsson, S. Riedel, and D. Yuret. 2007. The CoNLL 2007 shared task on dependency parsing. In *EMNLP-CoNLL*.

Sebastian Pado and Mirella Lapata. 2009. Cross-lingual annotation projection for semantic roles. *Journal of Artificial Intelligence Research*, 36:307–340.

Alexis Palmer and Caroline Sporleder. 2010. Evaluating FrameNet-style semantic parsing: the role of coverage gaps in FrameNet. In *COLING*.

M. Palmer, D. Gildea, and P. Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Hoifung Poon and Pedro Domingos. 2009. Unsupervised semantic parsing. In *EMNLP*.

Sameer Pradhan, Wayne Ward, and James H. Martin. 2008. Towards robust semantic role labeling. *Computational Linguistics*, 34:289–310.

Jason Rennie. 2001. Improving multi-class text classification with Naive bayes. Technical Report AITR-2001-004, MIT.

M. Sammons, V. Vydiswaran, T. Vieira, N. Johri, M. Chang, D. Goldwasser, V. Srikumar, G. Kundu, Y. Tu, K. Small, J. Rule, Q. Do, and D. Roth. 2009. Relation alignment for textual entailment recognition. In *Text Analysis Conference (TAC)*.

Dan Shen and Mirella Lapata. 2007. Using semantic roles to improve question answering. In *EMNLP*.

Richard Socher, Andrew Maas, and Christopher Manning. 2011. Spectral chinese restaurant processes: Nonparametric clustering based on similarities. In *AISTATS*.

Mihai Surdeanu, Adam Meyers Richard Johansson, Lluís Màrquez, and Joakim Nivre. 2008. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL 2008: Shared Task*.

Richard Swier and Suzanne Stevenson. 2004. Unsupervised semantic role labelling. In *EMNLP*.

Yee Whye Teh. 2010. Dirichlet processes. In *Encyclopedia of Machine Learning*. Springer.

Ivan Titov and Alexandre Klementiev. 2011. A Bayesian model for unsupervised semantic parsing. In *ACL*.

Ivan Titov and Mikhail Kozhevnikov. 2010. Bootstrapping semantic analyzers from non-contradictory texts. In *ACL*.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: A simple and general method for semi-supervised learning. In *ACL*.

Lonneke van der Plas, Paola Merlo, and James Henderson. 2011. Scaling up automatic cross-lingual semantic role annotation. In *ACL*.

Dekai Wu and Pascale Fung. 2009. Semantic roles for SMT: A hybrid two-pass model. In *NAACL*.

Dekai Wu, Marianna Apidianaki, Marine Carpuat, and Lucia Specia, editors. 2011. *Proc. of Fifth Workshop on Syntax, Semantics and Structure in Statistical Translation*. ACL.