Unsupervised Discovery of a Statistical Verb Lexicon

Trond Grenager and Christopher D. Manning
Computer Science Department
Stanford University
Stanford, CA 94305
{grenager, manning}@cs.stanford.edu

Abstract

This paper demonstrates how unsupervised techniques can be used to learn models of deep linguistic structure. Disambiguating the *semantic roles* of a verb's dependents is a necessary natural language understanding task, and the supervised approach, while successful, is inherently limited by the sparsity of observations in the training set. We present an unsupervised method for learning models of verb behavior directly from unannotated text corpora. Our learned lexicons are similar to resources such as VerbNet and PropBank, but include also statistics about the *linkings* allowed by each verb, and the head words in each role. Our method is based on a structured probabilistic model of the domain, and learning is performed with the EM algorithm. We evaluate performance in two ways. First, we evaluate the learned model as a semantic role labeler, relative to the PropBank annotation, and find that it reduces 30% of the precision error between an informed baseline and the best supervised system. Second, we inspect the linking models learned for each verb, and show that it learns the correct linking behavior.

1 Introduction

An important source of ambiguity that must be resolved by any natural language understanding system is the mapping between syntactic dependents of a predicate and the *semantic roles*[1] that they each express. The ambiguity stems from the fact that each predicate can allow several alternate mappings, or *linkings*[2], between its semantic roles and their syntactic realization. For example, the verb *increase* can be used in two ways:

(1) The Fed increased interest rates.
(2) Interest rates increased yesterday.

Syntactically, the instances are identical: they both have a noun phrase as the subject, and another noun phrase following the verb. However, while the subject of *increase* expresses the agent role in the first, it instead expresses the patient role in the second. Pairs of linkings such as this allowed by a single predicate are often called *diathesis alternations* (Levin, 1993).

[1] Also called *thematic roles*, *theta roles*, or *deep cases*.
[2] Often called *frames*.

The current state-of-the-art approach to resolving this ambiguity is to use discriminative classifiers, trained on hand-tagged data, to classify the semantic role of each dependent. A drawback of this approach is that the sparsity of evidence of the phenomena to be learned is extreme, even in a relatively large corpus. The two main hand-tagged corpora are PropBank and FrameNet, the former of which currently has broader coverage. However, even PropBank, which is based on the 1M word WSJ section of the Penn Treebank, is insufficient in quantity and genre to exhibit many things. A perfectly common verb like *flap* occurs only twice, across all morphological forms. The first example is an adjectival use *flapping wings*, and the second is a rare intransitive use with an agent argument and a path *ducks flapping over Washington*. From this data, one cannot learn the basic alternation pattern for *flap*: *the bird flapped its wings/the wings flapped*. Moreover, to automatically disambiguate linkings with identical syntactic realizations, one also needs a model of the words that can occur in each role. In PropBank, most of the verbs have been seen less than five times, hardly enough to form good estimates of the words occuring in each role. Without massive expansion of hand-built resources, supervised training data cannot be the basis of accurate models of verb linking behavior.

One way to address these challenges is to learn models of verb behavior directly from raw unannotated text, of which there is plenty. The models learned by our approach can be used discriminatively, to label parse tree constituents for semantic role. Perhaps more importantly, they provide a broad-coverage lexical resource which, for each verb type, provides statistics about the frequency with which it exhibits different linking patterns, and for each role of each verb statistics about the frequency of head words appearing in that role. This information may prove useful in other language processing tasks, such as parsing.

We apply our approach to the task of learning models of English verbs, which we then evaluate at two levels: at the level of verb tokens, we can evaluate our assigned semantic roles with respect to those given in an annotated corpus such as PropBank, and at the level of verb types, we can evaluate the success of the model in learning a verb's linking patterns, with respect to the PropBank frames.

Most existing work uses either existing verb lexicons or human annotation of semantic roles in texts. There has been work in using verb lexicons to replace supervised semantic role training data (Swier and Stevenson, 2004) and informative studies of particular cases, such as classifying intransitive/transitive patterns (Merlo and Stevenson, 2001), but we believe this is the first system that clusters unsupervised data so as to simultaneously learn verb classes, verb linking patterns, and verb role fillers.

Relation	Description
subj	NP preceding verb
np#n	NP in the nth position following verb
np	NP that is not the subject and not immediately following verb
cl#n	Complementary clause in the nth position following verb
cl	Complementary clause not immediately following verb
xcl#n	Complementary clause without subject in the nth position following verb
xcl	Complementary clause without subject not immediately following verb
acomp#n	Adjectival complement in the nth position following verb
acomp	Adjectival complement not immediately following verb
prep_x	Prepositional modifier with preposition x
advmod	Adverbial modifier
advcl	Adverbial clause

Table 1: The set of syntactic relations we use, where $n \in \{1, 2, 3\}$ and x is contained in a small set of English prepositions.

2 Learning Setting

Our goal is to learn a model which relates a verb, its semantic roles, and their possible syntactic realizations. As is the case with most semantic role labeling research, we do not attempt to model the syntax itself, and instead assume the existence of a syntactic parse of the sentence (Gildea and Jurafsky, 2002; Pradhan et al., 2005; Punyakanok et al., 2005). The parse may be from a human annotator, where available, or from an automatic parser. We can easily run our system on completely unannotated text by first running an automatic tokenizer, part-of-speech tagger, and parser to turn the text into tokenized, tagged sentences with associated parse trees. We also do not model word morphology (verb or noun), so in all cases we preprocess words to obtain their base form using a rule-based morphological analyzer based on Minnen et al. (2001).

In order to keep the model simple, and independent of any particular choice of syntactic representation, we use an abstract representation of syntax. We define a small set of *syntactic relations*, listed in Table 1, each of which describes a possible syntactic relationship between the verb and a dependent. Our goal was to choose a set that provides sufficient syntactic information for the semantic role decision, while remaining accurately computable from any reasonable parse tree using simple deterministic rules. Note that other sets would have been possible: as an example, our set does not include the relations *direct object* or *indirect object*, since this distinction is difficult to make accurately with rules alone. We obtain the syntactic relations used in our experiments by collapsing the set of typed dependencies that are returned by the Stanford Parser (http://nlp.stanford.edu/software/).

We also must choose a representation for semantic roles. We choose to allow each verb a small fixed number of roles, in the manner similar to PropBank's $ARG0 \ldots ARG5$. We also designate a single adjunct role which is shared by all verbs, similar to PropBank's $ARGM$ role. We say "similar" because our system never observes the PropBank roles (or any human annotated semantic roles) and so cannot possibly use the same names. Our system assigns arbitrary integer names to the roles that it discovers, just as clustering systems give arbitrary numbers to the clusters they discover. In practice, while our system is not guaranteed to choose role names that are consistent with PropBank, it often does anyway, which is a consequence of the constrained form of the linking model. This is described in more detail in Section 3.1.

Given the syntactic relations and semantic roles, we convert our parsed corpora into a simple format: a set of *verb instances*, each of which represents an occurrence of a verb in a sentence. A verb instance consists of the base form (lemma) of

Sentence: *Two years ago, a Canadian reader bet Omni Magazine $1,000 that it couldn't debunk the uncanny goings-on in "the Oregon Vortex," a former Indian burial ground in southern Oregon.*

Verb: bet		
Syntactic Relation	**Semantic Role**	**Head Word**
subj	ARG0	reader/NN
np#1	ARG2	Magazine/NNP
np#2	ARG1	$/$
cl#3	ARG3	debunk/VB

Verb: debunk		
Syntactic Relation	**Semantic Role**	**Head Word**
subj	ARG0	it/PRP
np#1	ARG1	going/NNS

Figure 1: An example sentence taken from the Penn Treebank (wsj_0413.mrg) and the two verb instances extracted from it. The semantic roles listed are taken from the PropBank annotation, but are not observed in the unsupervised training method.

the observed verb, and for each syntactic dependent of the verb, the dependent's syntactic relation, semantic role, and head word (including part of speech information). An example Penn Treebank sentence, and the verb instances extracted from it, are given in Figure 1. Though the system does not observe the semantic roles in training, we have shown the roles assigned by PropBank for illustration. These verb instances then serve as the input to our model, both in training and testing.

3 Probabilistic Model

Our learning method is based on a structured probabilistic model of the domain. The model represents a joint probability distribution of a verb instance, including the verb type, the particular linking used in the instance, and for each dependent of the verb, its syntactic relation to the verb, semantic role, and head word. Recall that a directed model such as ours can be viewed equivalently as a method for factoring a joint distribution over a large set of variables into a more tractable set of local distributions, or as a process for generating (sampling) a novel data instance. The latter is a more intuitive interpretation, so we present first the generative process that our model factorization corresponds to. The model is shown in graphical form in Figure 2.

We now show how the model can be used to generate a novel verb instance. We use as our running example the generation of the instance of the verb *increase* in the following sentence:

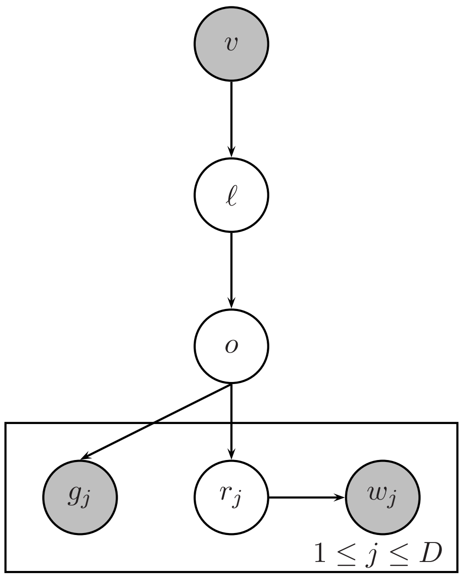
The Fed *increased* rates last month.

We begin by generating the verb lemma v, in this case *run*. Conditioned on the verb *run*, we next generate a linking ℓ, which defines both the set of core semantic roles to be expressed, as well as the syntactic relations that express them. In our example, we sample the transitive linking $\ell = \{ARG0 \rightarrow subj, ARG1 \rightarrow dobj\}$, a common one for the verb *increase* (the unaccusative intransitive $\{ARG1 \rightarrow subj\}$ is also allowed: *The rates increased.*). Conditioned on the linking, we next generate an *ordered* linking o, giving a final position in the dependent list for each role and relation in the linking ℓ, while also optionally inserting one or more adjunct roles. In our example, we generate the vector $o = [(ARG0, subj), (ARG1, np\#1), (ARGM, np\#2)]$. In doing so we've specified positions for $ARG0$ and $ARG1$, and added one adjunct role $ARGM$ in the third position. Note that the length of the ordered linking o is equal to the total number of dependents D of the verb instance.

Now we iterate through each of the dependents $1 \leq j \leq D$, generating each in turn. Because the semantic role r_j and syntactic relation g_j are completely determined by the ordered linking o, it remains only to sample the head word of each dependent, conditioned on the semantic role of that dependent. In this example, we generate the head words $w_1 = Fed/NNP$, $w_2 = rate/NNS$, and $w_3 = month/NN$. Note that we reduce our head words to their morphological base form and augment them with part of speech information.

Before defining the model more formally, we pause to justify some of the choices made in designing the model. First, we chose to distinguish between a verb's *core arguments* and its *adjuncts*. While core arguments must be associated with a semantic role that is verb specific (such as the patient role of *increase*: the *rates* in our example), adjuncts are generated by a role that is verb independent (such as the time of a generic event: *last month* in our example). Linkings include mappings only for the core semantic roles, resulting in a small, focused set of possible linkings for each verb. If, in contrast, one were to include adjuncts in the linkings, the set of possible linkings would be very large, since an almost unbounded number of temporal or locative modifiers can be added in a number of places in the dependent list.

A consequence of this choice is that we introduce uncertainty between the choice of linking and



Sentence: The Fed increased rates last month.

Verb: increase		
Syntactic Relation	**Semantic Role**	**Head Word**
subj	ARG0	Fed/NNP
np#1	ARG1	rates/NNS
np#2	ARGM	month/NN

$v =$ *increase*
$\ell = \{ARG0 \rightarrow subj, ARG1 \rightarrow np\#1\}$
$o = [(ARG0, subj),$
$(ARG1, np\#1),$
$(ARGM, np\#2)]$
$(g_1, r_1, w_1) = (subj, ARG0, Fed/NNP)$
$(g_2, r_2, w_2) = (np\#1, ARG1, rate/NNS)$
$(g_3, r_3, w_3) = (np\#2, ARGM, month/NN)$

Figure 2: A graphical representation of the verb linking model, with example values for each variable. The rectangle is a *plate*, indicating that the model contains multiple copies of the variables shown within it, in this case, one for each dependent j. Variables that are observed during learning are shaded. For clarity, we do not show variables for the model parameters.

its realization in the dependent list of a verb instance. The observed verb instance taken from the data does not explicitly model the distinction between core arguments and adjuncts. For example, if we observe an NP following a verb, it may be either a core role (an object of the verb, such as *rates* in the example above) or an adjunct (such as the temporal modifier *last month* in the example). Thus we need to consider the case in which the linking included only a mapping for the subject, and the NP was inserted as an adjunct. There are many ways to model this phenomenon; we chose to add the ordered linking variable o, which arranges the core arguments in the dependent list, optionally inserting adjunct roles with new syntactic relations.[3]

We now present the model formally as a factored joint probability distribution. We factor the joint probability distribution into a product of the probabilities of each instance:

$$\mathrm{P}(\mathcal{D}) = \prod_{i=1}^{N} \mathrm{P}(v^{(i)}, \ell^{(i)}, o^{(i)}, \mathbf{g}^{(i)}, \mathbf{r}^{(i)}, \mathbf{w}^{(i)}) \quad (1)$$

where we assume there are N instances, and we have used the vector notation $\mathbf{g}$ to indicate the vector of variables g_j for all values of j (and similarly for $\mathbf{r}$ and $\mathbf{w}$). We then factor the probability of each instance using the independencies shown in Figure 2 as follows:

$$\mathrm{P}(v, \ell, o, \mathbf{g}, \mathbf{r}, \mathbf{w}) = \mathrm{P}(v)\mathrm{P}(\ell|v)\mathrm{P}(o|\ell) \prod_{j=1}^{M} \mathrm{P}(g_j|o)\mathrm{P}(r_j|o)\mathrm{P}(w_j|r_j) \quad (2)$$

where we have assumed that there are M dependents of this instance.

We now describe more formally each of the conditional probability distributions given in Equation 2. The verb v is always observed in our data, so we don't formally define $\mathrm{P}(v)$. The probability of generating the linking given the verb $\mathrm{P}(\ell|v)$ is a multinomial over possible linkings. (Note, however, that the way in which we estimate this multinomial from data is complex, as is described in detail in the next section.) Next, the probability of a particular ordering of the linking $\mathrm{P}(o|\ell)$ is determined only by the number of adjuncts dependents that are added to o. One pays a constant penalty for each adjunct that is added to the dependent list, but otherwise all orderings of the roles are equally likely. Formally, this means that the ordering o is distributed according to the

[3] An alternative modeling choice would have been to add a state variable to each dependent, indicating which of the roles in the linking have been "used up" by previous dependents in the list. This would have equivalent representational power, but be more difficult to communicate.

Role	Linking Operations
ARG0	Add $ARG0$ to subj
ARG1	No operation Add $ARG1$ to np#1 Add $ARG1$ to cl#1 Add $ARG1$ to xcl#1 Add $ARG1$ to acomp#1 Add $ARG1$ to subj, replacing $ARG0$
ARG2	No operation Add $ARG2$ to prep_x, $\forall x$ Add $ARG2$ to np#1, shifting $ARG1$ to np#2
ARG3	No operation Add $ARG3$ to prep_x, $\forall x$
ARG4	No operation Add $ARG4$ to prep_x, $\forall x$

Table 2: The set of linking construction operations. To construct a linking, select one operation from each list.

geometric distribution of the difference between its length and the length of ℓ, with constant parameter λ.

Next, $\mathrm{P}(g_j|o)$ and $\mathrm{P}(r_j|o)$ are completed deterministic: the syntactic relation and semantic role for position j are specified in the ordering o. Finally, the word given the role $\mathrm{P}(w_j|r_j)$ is distributed as a multinomial over words.

3.1 Linking Model

The linking model uses a weak universal theory of verb argument realization. The additional structure is not shown in Figure 2 for simplicity, but instead is shown in Figure **??**. In designing the linking model, we intended to (1) constrain the set of linkings for each verb to a set of tractable size which are linguistically plausible[4], and (2) facilitate the construction of a structured prior distribution over this set, which gives higher weight to linkings that are known to be more common. A simple way to achieve both goals is to model the *derivation* of each linking from a *primitive root*, which we take as the simple transitive linking $\{ARG0 \rightarrow subj, ARG1 \rightarrow dobj\}$. This underlying form echoes the basic *Spec-Complement* structure of VP in Chomskyan theories, or Dowty's *proto-agent* and *proto-patient* roles

To derive a linking from the root, we go through two phases of modifications. In the first phase, we add as many as k new roles to the linking with *oblique* syntactic relations: those relations that are the object of a preposition. As an example, the linking $\{ARG0 \rightarrow subj, ARG1 \rightarrow dobj, ARG2 \rightarrow prep\_to\}$ can be derived from the transitive root by adding role $ARG2$ to the oblique relation $prep\_to$. Then, in the second step, we choose an *alternation* and use it to modify the arguments. For example, the *dative* alternation moves $ARG2$ from the relation $prep\_to$ to the relation $iobj$, resulting in the linking $\{ARG0 \rightarrow subj, ARG1 \rightarrow dobj, ARG2 \rightarrow iobj\}$. The full set of operations used at each derivation step is shown in Table 2, and the derivation process is shown as a graphical model in Figure **??**.

Given this derivation process, it is easy to created a structured prior: we just place *pseudocounts* on the operations that are likely *a priori* across all verbs.[5] Generally, we place high pseudocounts on the null operations (which preserve the basic transitive structure) and on the two ways of obtaining intransitive linkings. Note that the use of this structured prior has another desired side effect: it breaks the symmetry of the role names (because some linkings more likely than others) which encourages the model to adhere to canonical role naming conventions, at least for commonly occurring roles like $ARG0$ and $ARG1$.

The design of the linking model incorporates human prior knowledge about the structure of verb linkings and diathesis alternations. Indeed, the linking model provides a weak form of Universal Grammar, encoding the kinds of linking patterns that are known to occur in human languages. While not fully developed as a model of crosslinguistic verb argument realization, the model is not very English specific. It provides a not-very-constrained theory of alternations that captures common cross-linguistic patterns. Finally, though we do encode knowledge in the form of the model structure and associated prior distributions, we do not provide any verb-specific knowledge.

4 Learning

If we had fully observed data, our learning task would be trivial: compute the model parameters which maximize the training data, as follows.

$$\theta^* = \operatorname*{argmax}_\theta \mathrm{P}(\theta|\mathcal{I}) = \operatorname*{argmax}_\theta \prod_{i \in \mathcal{I}} \mathrm{P}(i;\theta)$$
$$= \operatorname*{argmax}_\theta \prod_{i \in \mathcal{I}} \mathrm{P}(v_i, \ell_i, \mathbf{p}_i, \mathbf{g}_i, \mathbf{r}_i, \mathbf{w}_i; \theta) \quad (3)$$

Because of the factorization of the joint distribution, this learning task would be trivial, com-

[4] The space of possible linkings is large: $O(|\mathcal{G}|^{|\mathcal{R}|})$.

[5] Formally, the parameters of the multinomial distributions from which the operations are sampled are distributed according to a Dirichlet distribution, and the pseudocounts are just parameters of the Dirichlet.

putable in closed form from relative frequency counts. Unfortunately, in our training set the variables ℓ_i, $\mathbf{p}_i$ and $\mathbf{r}_i$ are not observed, leaving us with a much harder optimization problem,

$$\begin{aligned}\theta^* &= \operatorname*{argmax}_\theta \prod_{i \in \mathcal{I}} \mathrm{P}(v_i, \mathbf{g}_i, \mathbf{w}_i; \theta) \\ &= \operatorname*{argmax}_\theta \prod_{i \in \mathcal{I}} \sum_{\ell_i, \mathbf{p}_i, \mathbf{r}_i} \mathrm{P}(v_i, \mathbf{g}_i, \mathbf{w}_i; \theta)\end{aligned} \tag{4}$$

In other words, we want model parameters which maximize the expected likelihood of the observed data, where the expectation is taken over the hidden variables for each instance.

The common solution to this problem is to use the Expectation-Maximization (EM) algorithm, a greedy search procedure over the parameter space which is guaranteed to increase the expected likelihood, and thus find a local maximum of the function. The EM algorithm is suitable when it is tractable to compute conditional distributions of the hidden variables given the model parameters (often called the E-step), and also to compute the maximum expected likelihood model parameters given conditional distributions over the hidden variables (often called the M-step). Unfortunately, in our case the M-step is possible but the E-step is intractable. It requires computing for each verb instance a distribution over the joint space of all linkings, position vectors, and roles for each dependent.

Fortunately, Monte Carlo methods present a simple solution: one can approximate an expectation over a set of variables by drawing samples from the underlying distribution and computing the weighted average over the sampled values instead of over all values.[6] One can incorporate this approach into the EM algorithm by replacing the conditional distributions over the hidden variables computed in the E-step with a set of samples over the hidden variables. Then, in the M-step, instead of computing the model parameters which maximize the expected likelihood, one computes model parameters which maximize the likelihood of the set of samples, as if they were observed data, which is easy. This algorithm is well known in the statistics literature, and is called Monte Carlo EM,

[6]A complete description of Gibbs and MCMC is beyond the scope of this paper; for a good introduction to MCMC techniques for machine learning, see Andrieu et al. (2003).

function LEARN($\mathcal{I}, N$)
 $\hat\theta \leftarrow$ INITIAL-VALUES()
 for each $i \in \mathcal{I}$
  $\mathbf{r}_i \leftarrow$ GET-DEFAULT-ROLES($\mathbf{g}_i$)
 repeat
  $S \leftarrow \emptyset$
  for each $i \in \mathcal{I}$
   for $k \leftarrow 1$ **to** N
    for each $j \in \mathcal{D}(i)$
     sample r_i^j **from** $\mathrm{P}(r_i^j | v_i, \mathbf{r}_i^{-j}, \mathbf{g}_i, \mathbf{w}_i; \theta)$
    $\ell_i \leftarrow$ BUILD-LINKING($\mathbf{r}_i, \mathbf{g}_i$)
   $S \leftarrow S \cup (\ell_i, \mathbf{r}_i, \mathbf{g}_i, \mathbf{w}_i)$
  SAMPLE-ROLE-CLASSES(S, θ)
  $\hat\theta_{-1} \leftarrow \hat\theta$
  $\hat\theta \leftarrow \operatorname{argmax}_\theta \mathrm{P}(S|\theta)$
 until $\mathrm{P}(S|\hat\theta) - \mathrm{P}(S|\hat\theta_{-1}) < \epsilon$
 return (θ)

Figure 3: Pseudocode for the basic learning algorithm, without role clustering. The call to SAMPLE-ROLE-CLASSES is optional, and the function is defined in Figure 4.

or Stochastic EM, depending on the precise variant (Celeux and Diebolt, 1986; Wei and Tanner, 1990).

Thus our problem is reduced to one of drawing samples from the distribution over hidden variable values, given values for the model parameters. We use Gibbs sampling, a Markov Chain Monte Carlo (MCMC) algorithm which is suitable for sampling from factored probabilities distributions (Geman and Geman, 1984) . In Gibbs sampling, one samples from a factored joint distribution by iteratively sampling each of the variables in turn, conditioned on the values fixed for the others, outputting complete samples at specified intervals. We apply this to our model by defining a procedure for iteratively sampling values for the hidden variables for each instance, as described in the next section.

The pseudocode for the complete unsupervised learning algorithm is given in Figure **??**. Note that it alternates between sampling hidden variables for each instance, and finding parameters θ which maximize the likelihood of these samples. In the inner loop we sample a new role r_i^j, conditioned on the current values of roles $\mathbf{r}_i^{-j}$ at all other dependents $j' \neq j$, as well as the observed syntactic relations $\mathbf{g}_i$ and words $\mathbf{w}_i$ for the instance. After sampling each role separately, we build the link-

function SAMPLE-ROLE-CLASSES(S, θ)
for each $v \in \mathcal{V}$
for each $r \in \mathcal{R}$
sample c_v^r **from** $\mathrm{P}(c_v^r|S;\theta)$

Figure 4: Pseudocode for the role class sampling function.

ing ℓ_i that is determined by the current choice of role, and accumulate the complete samples in the set S. The sampling distribution for r_i^j is easy to compute as a simple product of the model parameters:

$$\mathrm{P}(r_i^j|v_i, \mathbf{r}_i^{-j}, \mathbf{g}_i, \mathbf{w}_i; \theta) \\ \propto \mathrm{P}(\ell_i^*|v_i)\mathrm{P}(\mathbf{p}_i^*|\ell_i^*)\mathrm{P}(\mathbf{r}_i, \mathbf{g}_i|\ell_i^*, \mathbf{p}_i^*)\mathrm{P}(w_i^j|r_i^j, v)$$

where ℓ_i^* and $\mathbf{p}_i^*$ are the linking and position that are uniquely determined by the choice of r_i^j, given the values for the other variables.

5 Role Clustering

Another way to address the language sparsity issues inherent in learning a statistical verb lexicon is to share information between verb types. The statistical estimates in our lexicon which suffer the most from sparsity are the head word distributions for each role of each verb. For example, while a hundred verb instances may be enough to make a reasonable estimate of a verb's linking distribution, it is not enough to properly estimate the distribution over words that can occur as $ARG0$ for the verb. To address this issue, we augmented our basic model with role clusters.

In the basic model each word w_i^j is generated conditioned on both the verb v_i and the role r_i^j. In the role clustering model, we introduce a set of *role classes* $\mathcal{C}_r$ for each role $r \in \mathcal{R}$. Also, for each role r and verb v we introduce an associated class variable c_v^r which takes values from $\mathcal{C}_r$ indicating which class that role of that verb belongs to. Note that these class variables are generated for each role of each verb at the *type level*, not for each verb instance at the token level. Then, we generate the dependent head words from the role class, rather than from the verb and role itself. The resulting model then factorizes as follows:

$$\mathrm{P}(\mathcal{I}, \mathbf{c}) = \mathrm{P}(\mathbf{c}) \prod_{i \in \mathcal{I}} \mathrm{P}(v_i, \ell_i, \mathbf{p}_i, \mathbf{g}_i, \mathbf{r}_i, \mathbf{w}_i|\mathbf{c}) \quad (5)$$

where the first term is factored as the product of multinomials

$$\mathrm{P}(\mathbf{c}) = \prod_{c_v^r} \mathrm{P}(c_v^r) \quad (6)$$

and the second term is factored as

$$\mathrm{P}(v_i)\mathrm{P}(\ell_i|v_i)\mathrm{P}(\mathbf{p}_i|\ell_i)\mathrm{P}(\mathbf{g}_i, \mathbf{r}_i|\ell_i, \mathbf{p}_i) \cdot \\ \prod_{j \in D(i)} \mathrm{P}(w_i^j|c_{v_i}^{r_i^j}) \quad (7)$$

By adding these new unobserved role class variables we have complicated our inference task. Fortunately, we can easily modify our stochastic EM algorithm to accomodate the new variables: we sample their values during the E-step. In particular, we add it as a second stage of the E-step, so that we can condition on the samples of $\mathbf{r}_i$ and $\mathbf{w}_i$ accumulated during the first stage. The sampling distribution is then the following:

$$\mathrm{P}(c_v^r|S) \propto \mathrm{P}(c_v^r) \prod_{w_i^j \in S_v^r} \mathrm{P}(w_i^j|c_v^r) \quad (8)$$

where we uses the notation S_v^r to indicate the dependents of instances of the verb v from the set of samples that are labeled with role r.

6 Datasets and Evaluation

We evaluate the accuracy of the learned models at both the token and type level. In both cases we train our models with verb instances extracted from two parsed corpora: the Wall Street Journal (WSJ) section of the Penn Treebank (PTB), which was parsed by human annotators (Marcus et al., 1993), and the Brown Laboratory for Linguistic Information Processing (BLLIP) corpus of WSJ text, which was parsed automatically by the Charniak parser (Charniak, 2000). In all cases, when training a model, we specify a set of target verb types, and build our training set by adding a fixed number M of instances of each verb type. We first add the hand-parsed instances from the PTB, and if the number of training instances of that type is still less than M, we make up the difference by adding automatically-parsed instances from BLLIP. Note that even with the PTB and BLLIP combined, the number of training instances available for each verb type may be less than the target number M. Finally, we also add the test verb instances (if there are any) to the training set, as is customary in unsupervised learning.

For the token level evaluation, we use our system discriminatively to label the dependents of target verb instances for semantic role. As is customary in semantic role labeling, we use the sentences in PTB section 23 for testing, and PTB section 24 for development. All free parameters were tuned on the development set, and the test set was only used for final experiments. The test set consists of 4269 verb instances and 1099 different verb types. When training models for the semantic role labeling evaluation, we take these 1099 verbs as our set of target verbs for construction of the training set.

In testing, we use the system to label each of the dependents of test verb instances with a semantic role. Because we do not observe the gold standard semantic roles at training time, we do not know which of the gold semantic role labels each of our guessed labels corresponds to. Therefore we must choose an alignment between the guessed labels and the gold labels once for each verb type. We do so optimistically, by choosing the gold label for each guessed label which maximizes the number of correct guesses. This is a well known approach to evaluation in unsupervised learning: when it is used to compute accuracy, the resulting metric is sometimes called *cluster purity*. While this amounts to "peeking" at the answers before evaluation, the amount of human knowledge that is given to the system is small: it corresponds to the effort required to hand assign a "name" to each label that the system proposes. Instead of computing accuracy conditioned on the alignment, we compute precision, recall, and F1 measure, to be consistent with the metrics reported in the supervised semantic role labeling literature.

For the type level evaluation, we sampled 20 verbs at random from the PropBank frames files and compared linkings in the the PropBank lexicon entry with the linking distribution from a model learned using our unsupervised technique.

7 Results

For the token-level evaluation, we report results for labeling the dependents of the verb instances in the test set for semantic role. As is customary, we report results both for the task of distinguishing between all roles (including core and adjunct), and for the task of distinguishing only among the core roles (in which the adjunct role is ignored). The availability of syntactic relations in our system allows us to use an informed baseline, in which the syntactic relation itself is used as the semantic role label. The syntactic relations *subj*, *dobj*, and *iobj* are all assumed to be core semantic roles, and all other relations are assumed to be adjunct roles. This baseline method gives an F1 of 0.677 on the all roles task, and 0.807 on the core roles task. Note the disparity between precision and recall: this is a consequence of conditioning our labeling on syntactic relations that are extracted from the original parse using a set of high-precision, lower-recall rules. To get a sense of the restrictiveness of the approach, we compute an upper bound by defining a method which uses an oracle to guess the correct role for each of the dependents proposed by the syntactic relations rules. This yields a precision of 0.966 and a recall of 0.691, indicating that the current syntactic relation extraction system does impose a significant limitation on recall.

	All Roles			**Core Roles**		
	P	R	F1	P	R	F1
Baseline	.811	.580	.677	.927	.714	.807
Test Only						
+100 Train						
+200						
+400						
+800						
Up Bnd.	.966	.691	.806			

Table 3: Summary of experimental results on labeling all verb instances in Section 23 for semantic role. Performance measures are microaveraged over all instances of all verbs.

We have listed the semantic role labeling performance of the unsupervised system for various amounts of training data.

At the type level,

8 Conclusion

We apply unsupervised learning to the problem of learning a model relating deep semantic argument structure of verbs to their surface syntax. We define a generative probabilistic model of the domain, as well as an algorithm for training such models from unsupervised data. We then evaluate the performance of our learned models against gold labels from PropBank. While the performance of the models learned in this manner is not comparable with those learned using supervised techniques, it achieves a cluster purity of 0.876, reducing error over the baseline by 39%. In future work we would like to use unsupervised training to improve the performance of supervised models on tasks such as semantic role labeling and even parsing. We would also like to develop models which can generalize across verbs, and use patterns observed for common verbs to properly label roles for rare verbs.

References

C. Andrieu, N. d Freitas, A. Doucet, and M. I. Jordan. 2003. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43.

G. Celeux and J. Diebolt. 1986. The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics Quarterly*, 2:73–82.

E. Charniak. 2000. A maximum entropy inspired parser. In *NAACL 2002*.

S. Geman and D. Geman. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transitions on Pattern Analysis and Machine Intelligence*, 6:721–741.

D. Gildea and D. Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28.

B. Levin. 1993. *English Verb Classes and Alternations*. University of Chicago Press.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. 1993. Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19:313–330.

P. Merlo and S. Stevenson. 2001. Automatic verb classification based on statistical distributions of argument structure. *Computational Linguistics*, 27(3):373–408.

G. Minnen, J. Carroll, and D. Pearce. 2001. Applied morphological processing of english. *Natural Language Engineering*, 7:207–223.

S. Pradhan, W. Ward, K. Hacioglu, J. Martin, and D. Jurafsky. 2005. Semantic role labeling using different syntactic views. In *ACL 2005*.

V. Punyakanok, D. Roth, and W. Yih. 2005. Generalized inference with multiple semantic role labeling systems shared task paper. In *CoNLL 2005*.

R. S. Swier and S. Stevenson. 2004. Unsupervised semantic role labeling. In *EMNLP 2004*.

G. C. G. Wei and M. A. Tanner. 1990. A Monte Carlo implementation of the EM algorithm and the poor man's data augmentation algorithm. *Journal of the American Statistical Association*, 85:699–704.