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. Determining the *semantic roles* of a verb's dependents is an important step in natural language understanding. We present a method for learning models of verb argument patterns directly from unannotated text. The learned models are similar to existing verb lexicons such as VerbNet and PropBank, but additionally include statistics about the *linkings* used by each verb. The method is based on a structured probabilistic model of the domain, and unsupervised learning is performed with the EM algorithm. The learned models can also be used discriminatively as semantic role labelers, and when evaluated relative to the PropBank annotation, the best learned model reduces 28% of the error between an informed baseline and an oracle upper bound.

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.

The instances have apparently similar surface syntax: they both have a subject and a noun phrase directly 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).

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 (Gildea and Jurafsky, 2002; Pradhan et al., 2005; Punyakanok et al., 2005). A drawback of this approach is that even a relatively large training corpus exhibits considerable sparsity of evidence. The two main hand-tagged corpora are PropBank (Palmer et al., 2003) and FrameNet (Baker et al., 1998), 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* vs. *the wings flapped*.

We propose to address the challenge of data sparsity by learning models of verb behavior directly from raw unannotated text, of which there is plenty. This has the added advantage of being easily extendible to novel text genres and languages, and the possibility of shedding light on the question of human language acquisition. The models learned by our unsupervised approach provide a new broad-coverage lexical resource which gives statistics about verb behavior, information that may prove useful in other language processing tasks, such as parsing. Moreover, they may be used discriminatively to label novel verb instances for semantic role. Thus we evaluate them both in terms of the verb alternations that they learn and their accuracy as semantic role labelers.

This work bears some similarity to the substantial literature on automatic subcategorization frame acquisition (see, e.g., Manning (1993), Briscoe and Carroll (1997), and Korhonen (2002)). However, that research is focused on acquiring verbs' syntactic behavior, and we are focused on the acquisition of verbs' linking behavior. More relevant is the work of McCarthy and

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

Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing (EMNLP 2006), pages 1–8,
Sydney, July 2006.

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	Complement clause in the nth position following verb
cl	Complement clause not immediately following verb
xcl#n	Complement clause without subject in the nth position following verb
xcl	Complement 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 a preposition.

Korhonen (1998), which used a statistical model to identify verb alternations, relying on an existing taxonomy of possible alternations, as well as Lapata (1999), which searched a large corpus to find evidence of two particular verb alternations. There has also been some work on both clustering and supervised classification of verbs based on their alternation behavior (Stevenson and Merlo, 1999; Schulte im Walde, 2000; Merlo and Stevenson, 2001). Finally, Swier and Stevenson (2004) perform unsupervised semantic role labeling by using hand-crafted verb lexicons to replace supervised semantic role training data. However, we believe this is the first system to simultaneously discover verb roles and verb linking patterns from unsupervised data using a unified probabilistic model.

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. 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.

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. Our set does not include the relations *direct object* or *indirect object*, since this distinction can not be made deterministically on the basis of syntactic structure alone; instead, we opted to number the noun phrase (*np*), complement clause (*cl, xcl*), and adjectival complements (*acomp*) appearing in an unbroken sequence directly after the verb, since this is sufficient to capture the necessary syntactic information. The syntactic relations used in our experiments are computed from the typed dependencies returned by the Stanford Parser (Klein and Manning, 2003).

Sentence: *A deeper market plunge today could give them their first test.*

Verb: give		
Syntactic Relation	**Semantic Role**	**Head Word**
subj	ARG0	plunge/NN
np	ARGM	today/NN
np#1	ARG2	they/PRP
np#2	ARG1	test/NN

$$
\begin{aligned}
&v = give\\
&\ell = \{ARG0 \to subj, ARG1 \to np\#2\\
&\quad ARG2 \to np\#1\}\\
&o = [(ARG0, subj), (ARGM, ?),\\
&\quad (ARG2, np\#1), (ARG1, np\#2)]\\
&(g_1, r_1, w_1) = (subj, ARG0, plunge/NN)\\
&(g_2, r_2, w_2) = (np, ARG0, today/NN)\\
&(g_3, r_3, w_3) = (np\#1, ARG2, they/PRP)\\
&(g_4, r_4, w_4) = (np\#2, ARG1, test/NN)
\end{aligned}
$$

Figure 1: An example sentence taken from the Penn Treebank (wsj_2417), the verb instance extracted from it, and the values of the model variables for this instance. The semantic roles listed are taken from the PropBank annotation, but are not observed in the unsupervised training method.

We also must choose a representation for semantic roles. We 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 it discovers, just as clustering systems give

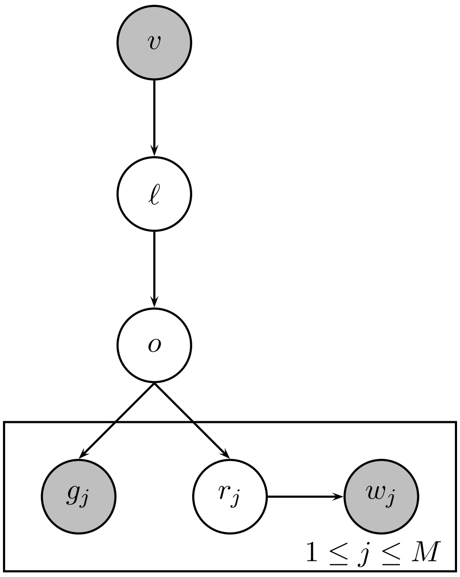

Figure 2: A graphical representation of the verb linking model, with example values for each variable. The rectangle is a *plate*, indicating that the model contains multiple copies of the variables shown within it: in this case, one for each dependent j. Variables observed during learning are shaded.

arbitrary names to the clusters they discover.[3]

Given these definitions, 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 the observed verb, and for each dependent of the verb, the dependent's syntactic relation and head word (represented as the base form with part of speech information). An example Penn Treebank sentence, and the verb instances extracted from it, are given in Figure 1.

3 Probabilistic Model

Our learning method is based on a structured probabilistic model of the domain. A graphical representation of the model is shown in Figure 2. The model encodes a joint probability distribution over the elements of a single verb instance, including the verb type, the particular linking, and for each dependent of the verb, its syntactic relation to the verb, semantic role, and head word.

We begin by describing the generative process to which our model corresponds, using as our running example the instance of the verb *give* shown in Figure 1. We begin by generating the verb lemma v, in this case *give*. Conditioned on the choice of verb *give*, 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 ditransitive linking $\ell = \{ARG0 \rightarrow subj, ARG1 \rightarrow np\#2, ARG2 \rightarrow np\#1\}$. Conditioned on this choice of 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 $o = [(ARG0, subj), (ARGM, ?), (ARG2, np\#1), (ARG1, np\#2)]$. In doing so we've specified positions for $ARG0$, $ARG1$, and $ARG2$ and added one adjunct role $ARGM$ in the second position. Note that the length of the ordered linking o is equal to the total number of dependents M of the verb instance. Now we iterate through each of the dependents $1 \leq j \leq M$, generating each in turn. For the core arguments, the semantic role r_j and syntactic relation g_j are completely determined by the ordered linking o, so it remains only to sample the syntactic relation for the adjunct role: here we sample $g_2 = np$. We finish by sampling the head word of each dependent, conditioned on the semantic role of that dependent. In this example, we generate the head words $w_1 = plunge/NN$, $w_2 = today/NN$, $w_3 = they/NN$, and $w_4 = test/NN$.

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. A consequence of this choice is that we introduce uncertainty between the choice of linking and its realization in the dependent list, which we represent with ordered linking variable o.[4]

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

[3] 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.

[4] 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.

probabilities of each instance:

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

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:

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

where we have assumed that there are M dependents of this instance. The verb v is always observed in our data, so we don't need to define $\mathrm{P}(v)$. The probability of generating the linking given the verb $\mathrm{P}(\ell|v)$ is a multinomial over possible linkings.[5] Next, the probability of a particular ordering of the linking $\mathrm{P}(o|\ell)$ is determined only by the number of adjunct 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, the ordering o is distributed according to the geometric distribution of the difference between its length and the length of ℓ, with constant parameter λ.[6] Next, $\mathrm{P}(g_j|o)$ and $\mathrm{P}(r_j|o)$ are completely deterministic for core roles: the syntactic relation and semantic role for position j are specified in the ordering o. For adjunct roles, we generate g_j from a multinomial over syntactic relations. Finally, the word given the role $\mathrm{P}(w_j|r_j)$ is distributed as a multinomial over words.

To allow for labeling elements of verb instances (verb types, syntactic relations, and head words) at test time that were unobserved in the training set, we must smooth our learned distributions. We use Bayesian smoothing: all of the learned distributions are multinomials, so we add *psuedocounts*, a generalization of the well-known *add-one smoothing* technique. Formally, this corresponds to a Bayesian model in which the parameters of these multinomial distributions are themselves random variables, distributed according to a Dirichlet distribution.[7]

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 Add $ARG2$ to np#1, shifting $ARG1$ to prep_with
ARG3	No operation Add $ARG3$ to prep_x, $\forall x$ Add $ARG3$ to cl#n, $1 < n < 3$
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.

3.1 Linking Model

The most straightforward choice of a distribution for $\mathrm{P}(\ell|v)$ would be a multinomial over all possible linkings. There are two problems with this simple implementation, both stemming from the fact that the space of possible linkings is large (there are $O(|\mathcal{G}+1|^{|\mathcal{R}|})$, where $\mathcal{G}$ is the set of syntactic relations and $\mathcal{R}$ is the set of semantic roles). First, most learning algorithms become intractable when they are required to represent uncertainty over such a large space. Second, the large space of linkings yields a large space of possible models, making learning more difficult.

As a consequence, we have two objectives when designing $\mathrm{P}(\ell|v)$: (1) constrain the set of linkings for each verb to a set of tractable size which are linguistically plausible, 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. Our solution is to model the *derivation* of each linking as a sequence of *construction operations*, an idea which is similar in spirit to that used by Eisner (2001). Each operation adds a new role to the linking, possibly replacing or displacing one of the existing roles. The complete list of linking operations is given in Table 2. To build a linking we select one operation from each list; the presence of a no-operation for each role means that a linking doesn't have to include all roles. Note that this linking derivation process is not shown in Figure 2, since it is possi-

[5] The way in which we estimate this multinomial from data is more complex, and is described in the next section.

[6] While this may seem simplistic, recall that all of the important ordering information is captured by the syntactic relations.

[7] For a more detailed presentation of Bayesian methods, see Gelman et al. (2003).

ble to compile the resulting distribution over linkings into the simpler multinomial $\mathrm{P}(\ell|v)$.

More formally, we factor $\mathrm{P}(\ell|v)$ as follows, where $\mathbf{c}$ is the vector of construction operations used to build ℓ:

$$\begin{aligned}\mathrm{P}(\ell|v) &= \sum_{\mathbf{c}} \mathrm{P}(\ell|\mathbf{c})\mathrm{P}(\mathbf{c}|v) \\ &= \sum_{\mathbf{c}} \prod_{i=1}^{|\mathcal{R}|} \mathrm{P}(c_i|v)\end{aligned}$$

Note that in the second step we drop the term $\mathrm{P}(\ell|\mathbf{c})$ since it is always 1 (a sequence of operations leads deterministically to a linking).

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. We place high pseudocounts on the no-operations (which preserve simple intransitive and transitive structure) and low pseudocounts on all the rest. 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 does incorporate 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 cross-linguistic 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, note that we do not provide any verb-specific knowledge; this is left to the learning algorithm.

4 Learning

Our goal in learning is to find parameter settings of our model which are likely given the data. Using θ to represent the vector of all model parameters, if our data were fully observed, we could express our learning problem as

$$\begin{aligned}\theta^* &= \operatorname*{argmax}_\theta \mathrm{P}(\theta|\mathcal{D}) = \operatorname*{argmax}_\theta \prod_{i=1}^{N} \mathrm{P}(d^i;\theta) \\ &= \operatorname*{argmax}_\theta \prod_{i=1}^{N} \mathrm{P}(v^i,\ell^i,o^i,\mathbf{g}^i,\mathbf{r}^i,\mathbf{w}^i;\theta)\end{aligned}$$

Because of the factorization of the joint distribution, this learning task would be trivial, computable in closed form from relative frequency counts. Unfortunately, in our training set the variables ℓ, o and $\mathbf{r}$ are hidden (not observed), leaving us with a much harder optimization problem:

$$\begin{aligned}\theta^* &= \operatorname*{argmax}_\theta \prod_{i=0}^{N} \mathrm{P}(v^i,\mathbf{g}^i,\mathbf{w}^i;\theta) \\ &= \operatorname*{argmax}_\theta \prod_{i=0}^{N} \sum_{\ell^i,o^i,\mathbf{r}^i} \mathrm{P}(v^i,\ell^i,o^i,\mathbf{g}^i,\mathbf{r}^i,\mathbf{w}^i;\theta)\end{aligned}$$

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. Although it is intractable to find exact solutions to optimization problems of this form, the Expectation-Maximization (EM) algorithm is 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.

While the M-step is clearly trivial, the E-step at first looks more complex: there are three hidden variables for each instance, $\ell, o,$ and $\mathbf{r}$, each of which can take an exponential number of values. Note however, that conditioned on the observed set of syntactic relations $\mathbf{g}$, the variables ℓ and o are completely determined by a choice of roles $\mathbf{r}$ for each dependent. So to represent uncertainty over these variables, we need only to represent a distribution over possible role vectors $\mathbf{r}$. Though in the worst case the set of possible role vectors is still exponential, we only need role vectors that are consistent with both the observed list of syntactic relations and a linking that can be generated by the construction operations. Empirically the number of linkings is small (less than 50) for each of the observed instances in our data sets.

Then for each instance we construct a conditional probability distribution over this set, which is computable in terms of the model parameters:

$$\mathrm{P}(\mathbf{r}, \ell_{\mathbf{r}}, o_{\mathbf{r}}, |v, \mathbf{g}, \mathbf{w}) \propto$$

$$\mathrm{P}(\ell_{\mathbf{r}}|v)\mathrm{P}(o_{\mathbf{r}}|\ell_{\mathbf{r}}) \prod_{j=1}^{M} \mathrm{P}(g_j|o_{\mathbf{r}})\mathrm{P}(r_j|o_{\mathbf{r}})\mathrm{P}(w_j|r_j)$$

We have denoted as $\ell_{\mathbf{r}}$ and $o_{\mathbf{r}}$ the values of ℓ and o that are determined by each choice of $\mathbf{r}$.

To make EM work, there are a few additional subtleties. First, because EM is a hill-climbing algorithm, we must initialize it to a point in parameter space with slope (and without symmetries). We do so by adding a small amount of noise: for each dependent of each verb, we add a fractional count of 10^{-6} to the word distribution of a semantic role selected at random. Second, we must choose when to stop EM: we run until the relative change in data log likelihood is less than 10^{-4}.

A separate but important question is how well EM works for finding "good" models in the space of possible parameter settings. "Good" models are ones which list linkings for each verb that correspond to linguists' judgments about verb linking behavior. Recall that EM is guaranteed only to find a local maximum of the data likelihood function. There are two reasons why a particular maximum might not be a "good" model. First, because it is a greedy procedure, EM might get stuck in local maxima, and be unable to find other points in the space that have much higher data likelihood. We take the traditional approach to this problem, which is to use random restarts; however empirically there is very little variance over runs. A deeper problem is that data likelihood may not correspond well to a linguist's assessment of model quality. As evidence that this is not the case, we have observed a strong correlation between data log likelihood and labeling accuracy.

5 Datasets and Evaluation

We train our models with verb instances extracted from three parsed corpora: (1) the Wall Street Journal section of the Penn Treebank (PTB), which was parsed by human annotators (Marcus et al., 1993), (2) the Brown Laboratory for Linguistic Information Processing corpus of Wall Street Journal text (BLLIP), which was parsed automatically by the Charniak parser (Charniak, 2000), and (3) the Gigaword corpus of raw newswire text (GW), which we parsed ourselves with the Stanford parser. In all cases, when training a model, we specify a set of target verb types (e.g., the ones in the test set), and build a training set by adding a fixed number of instances of each verb type from the PTB, BLLIP, and GW data sets, in that order.

	Coarse Roles			**Core Roles**		
Sec. 23	**P**	**R**	**F1**	**P**	**R**	**F1**
ID Only	.957	.802	.873	.944	.843	.891
CL Only						
Baseline	.856	.856	.856	.975	.820	.886
PTB Tr.	.889	.889	.889	.928	.898	.911
1000 Tr.	.897	.897	.897	.947	.898	.920
ID+CL						
Baseline	.819	.686	.747	.920	.691	.789
PTB Tr.	.851	.712	.776	.876	.757	.812
1000 Tr.	.859	.719	.783	.894	.757	.820
Sec. 24	**P**	**R**	**F1**	**P**	**R**	**F1**
ID Only	.954	.788	.863	.941	.825	.879
CL Only						
Baseline	.844	.844	.844	.980	.810	.882
PTB Tr.	.893	.893	.893	.940	.903	.920
1000 Tr.	.899	.899	.899	.956	.898	.925
ID+CL						
Baseline	.804	.665	.729	.922	.668	.775
PTB Tr.	.852	.704	.771	.885	.745	.809
1000 Tr.	.858	.709	.776	.900	.741	.813

Table 3: Summary of results on labeling verb instances in PropBank Section 23 and Section 24 for semantic role. Learned results are averaged over 5 runs.

For the semantic role labeling evaluation, we use our system to label the dependents of unseen verb instances for semantic role. We use the sentences in PTB section 23 for testing, and PTB section 24 for development. The development set consists of 2507 verb instances and 833 different verb types, and the test set consists of 4269 verb instances and 1099 different verb types. Free parameters were tuned on the development set, and the test set was only used for final experiments.

Because we do not observe the gold standard semantic roles at training time, we must choose an alignment between the guessed labels and the gold labels. 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.

As is customary, we divide the problem into two subtasks: *identification* (ID) and *classification* (CL). In the identification task, we identify the set of constituents which fill some role for a

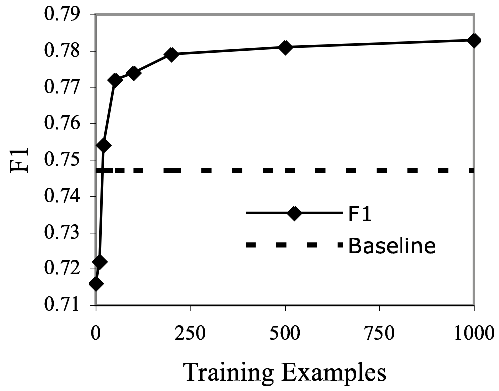

Figure 3: Test set F1 as a function of training set size.

target verb: in our system we use simple rules to extract dependents of the target verb and their grammatical relations. In the classification task, the identified constituents are labeled for their semantic role by the learned probabilistic model. We report results on two variants of the basic classification task: *coarse roles*, in which all of the adjunct roles are collapsed to a single $ARGM$ role (Toutanova, 2005), and *core roles*, in which we evaluate performance on the core semantic roles only (thus collapsing the $ARGM$ and unlabeled categories). We do not report results on the *all roles* task, since our current model does not distinguish between different types of adjunct roles. For each task we report precision, recall, and F1.

6 Results

The semantic role labeling results are summarized in Table 3. Our performance on the identification task is high precision but low recall, as one would expect from a rule-based system. The recall errors stem from constituents which are considered to fill roles by PropBank, but which are not identified as dependents by the extraction rules (such as those external to the verb phrase). The precision errors stem from dependents which are found by the rules, but are not marked by PropBank (such as the expletive "it").

In the classification task, we compare our system to an informed baseline, which is computed by labeling each dependent with a role that is a deterministic function of its syntactic relation. The syntactic relation *subj* is assumed to be $ARG0$, and the syntactic relations *np#1*, *cl#1*, *xcl#1*, and *acomp#1* are mapped to role $ARG1$, and all other dependents are mapped to $ARGM$.

Our best system, trained with 1000 verb instances per verb type (where available), gets an F1 of 0.897 on the coarse roles classification task on the test set (or 0.783 on the combined identification and classification task), compared with an F1 of 0.856 for the baseline (or 0.747 on the combined task), thus reducing 28.5% of the relative error. Similarly, this system reduces 35% of the error on the coarse roles task on development set.

Verb (△ F1)	**Learned Linkings**	
give	.57	{0=subj,1=np#2,2=np#1}
(+.436)	.24	{0=subj,1=np#1}
	.13	{0=subj,1=np#1,2=to}
work	.45	{0=subj}
(+.206)	.09	{0=subj,2=with}
	.09	{0=subj,2=for}
	.09	{0=subj,2=on}
pay	.47	{0=subj,1=np#1}
(+.178)	.21	{0=subj,1=np#1,2=for}
	.10	{0=subj}
	.07	{0=subj,1=np#2,2=np#1}
look	.28	{0=subj}
(+.170)	.18	{0=subj,2=at}
	.16	{0=subj,2=for}
rise	.25	{0=subj,1=np#1,2=to}
(+.160)	.17	{0=subj,1=np#1}
	.14	{0=subj,2=to}
	.12	{0=subj,1=np#1,2=to,3=from}

Table 4: Learned linking models for the most improved verbs. To conserve space, $ARG0$ is abbreviated as 0, and *prep_to* is abbreviated as *to*.

To get a better sense of what is and is not being learned by the model, we compare the performance of individual verbs in both the baseline system and our best learned system. For this analysis, we have restricted focus to verbs for which there are at least 10 evaluation examples, to yield a reliable estimate of performance. Of these, 27 verbs have increased F1 measure, 17 are unchanged, and 8 verbs have decreased F1. We show learned linkings for the 5 verbs which are most and least improved in Tables 4 and 5.

The improvement in the verb *give* comes from the model's learning the ditransitive alternation. The improvements in *work*, *pay*, and *look* stem from the model's recognition that the oblique dependents are generated by a core semantic role. Unfortunately, in some cases it lumps different roles together, so the gains are not as large as they could be. The reason for this conservatism is the relatively high level of smoothing in the word distribution relative to the linking distribution. These smoothing parameters, set to optimize performance on the development set, prevent errors of spurious role formation on other verbs. The improvement in the verb *rise* stems from the model correctly assigning separate roles each for the amount risen, the source, and the destination.

Verb **($\triangle$ F1)**	**Learned Linkings**	
help	.52	{0=subj,1=cl#1}
($-$.039)	.25	{0=subj,1=xcl#1}
	.16	{0=subj,1=np#1}
follow	.81	{0=subj,1=np#1}
($-$.056)	.13	{0=subj,1=cl#1}
make	.64	{0=subj,1=np#1}
($-$.133)	.23	{0=subj,1=cl#1}
leave	.57	{0=subj,1=np#1}
($-$.138)	.18	{0=subj}
	.12	{0=subj,1=cl#1}
close	.24	{0=subj,2=in,3=at}
($-$.400)	.18	{0=subj,3=at}
	.11	{0=subj,2=in}
	.10	{0=subj,1=np#1,2=in,3=at}

Table 5: Learned linking models for the least improved verbs. To conserve space, $ARG0$ is abbreviated as 0, and *prep_to* is abbreviated as *to*.

The poor performance on the verb *close* stems from its idiosyncratic usage in the WSJ corpus; a typical use is *In national trading, SFE shares closed yesterday at 31.25 cents a share, up 6.25 cents* (wsj_0229). Our unsupervised system finds that the best explanation of this frequent use pattern is to give special roles to the temporal (*yesterday*), locative (*at 31.25 cents*), and manner (*in trading*) modifiers, none of which are recognized as roles by PropBank. The decrease in performance on *leave* stems from its inability to distinguish between its two common senses (*left Mary with the gift* vs. *left Mary alone*), and the fact that PropBank tags Mary as $ARG1$ in the first instance, but $ARG2$ (beneficiary) in the second. The errors in *make* and *help* result from the fact that in a phrase like *make them unhappy* the Penn Treebank chooses to wrap *them unhappy* in a single S, so that our rules show only a single dependent following the verb: a complement clause (cl#1) with head word *unhappy*. Unfortunately, our system calls this clause $ARG1$ (omplement clauses following the verb are usually $ARG1$), but PropBank calls it $ARG2$. The errors in the verb *follow* also stem from a sense confusion: *the second followed the first* vs. *he followed the principles*.

7 Conclusion

We have demonstrated that it is possible to learn a statistical model of verb semantic argument structure directly from unannotated text. More work needs to be done to resolve particular classes of errors; for example, the one reported above for the verb *work*. It is perhaps understandable that the dependents occurring in the obliques *with* and *for* are put in the same role (the head words should refer to *people*), but it is harder to accept that dependents occurring in the oblique *on* are also grouped into the same role (the head words of these should refer to *tasks*). It seems plausible that measures to combat word sparsity might help to differentiate these roles: backing-off to word classes, or even just training with much more data. Nevertheless, semantic role labeling performance improvements demonstrate that on average the technique is learning verb linking models that are correct.

References

C. F. Baker, C. J. Fillmore, and J. B. Lowe. 1998. The Berkeley FrameNet project. In *ACL 1998*, pages 86–90.

T. Briscoe and J. Carroll. 1997. Automatic extraction of subcategorization from corpora. In *Applied NLP 1997*, pages 356–363.

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

J. M. Eisner. 2001. *Smoothing a probabilistic lexicon via syntactic transformations*. Ph.D. thesis, University of Pennsylvania.

A. Gelman, J. B. Carlin, H. S. Stern, and Donald D. B. Rubin. 2003. *Bayesian Data Analysis*. Chapman & Hall.

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

D. Klein and C. Manning. 2003. Accurate unlexicalized parsing. In *ACL 2003*.

A. Korhonen. 2002. *Subcategorization acquisition*. Ph.D. thesis, University of Cambridge.

M. Lapata. 1999. Acquiring lexical generalizations from corpora: A case study for diathesis alternations. In *ACL 1999*, pages 397–404.

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

C. D. Manning. 1993. Automatic acquisition of a large subcategorization dictionary. In *ACL 1993*, pages 235–242.

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.

C. McCarthy and A. Korhonen. 1998. Detecting verbal participation in diathesis alternations. In *ACL 1998*, pages 1493–1495.

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

M. Palmer, D. Gildea, and P. Kingsbury. 2003. The proposition bank: An annotated corpus of semantic roles. *Computational Linguistics*.

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*.

S. Schulte im Walde. 2000. Clustering verbs automatically according to their alternation behavior. In *ACL 2000*, pages 747–753.

S. Stevenson and P. Merlo. 1999. Automatic verb classification using distributions of grammatical features. In *EACL 1999*, pages 45–52.

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

K. Toutanova. 2005. *Effective statistical models for syntactic and semantic disambiguation*. Ph.D. thesis, Stanford University.