Unsupervised Induction of a Syntax-Semantics Lexicon Using Iterative Refinement

Hagen Fürstenau
CCLS
Columbia University
New York, NY, USA
hagen@ccls.columbia.edu

Owen Rambow
CCLS
Columbia University
New York, NY, USA
rambow@ccls.columbia.edu

Abstract

We present a method for learning syntax-semantics mappings for verbs from unannotated corpora. We learn *linkings*, i.e., mappings from the syntactic arguments and adjuncts of a verb to its semantic roles. By learning such linkings, we do not need to model individual semantic roles independently of one another, and we can exploit the relation between different mappings for the same verb, or between mappings for different verbs. We present an evaluation on a standard test set for semantic role labeling.

1 Introduction

A verb can have several ways of mapping its semantic arguments to syntax ("diathesis alternations"):

(1) a. We increased the response rate with SHK.
b. SHK increased the response rate.
c. The response rate increased.

The subject of *increase* can be the agent (1a), the instrument (1b), or the theme (what is being increased) (1c). Other verbs that show this pattern include *break* or *melt*.

Much theoretical and lexicographic (descriptive) work has been devoted to determining how verbs map their lexical predicate-argument structure to syntactic arguments (Burzio, 1986; Levin, 1993). The last decades have seen a surge in activity on the computational front, spurred in part by efforts to annotate large corpora for lexical semantics (Baker et al., 1998; Palmer et al., 2005). Initially, we have seen computational efforts devoted to finding classes of verbs that share similar syntax-semantics mappings from annotated and unannotated corpora (Lapata and Brew, 1999; Merlo and Stevenson, 2001). More recently, there has been an explosion of interest in semantic role labeling (with too many recent publications to cite).

In this paper, we explore learning syntax-semantics mappings for verbs from unannotated corpora. We are specifically interested in learning *linkings*. A linking is a mapping for one verb from its syntactic arguments and adjuncts to *all* of its semantic roles, so that individual semantic roles are not modeled independently of one another and so that we can exploit the relation between different mappings for the same verb (as in (1) above), or between mappings for different verbs. We therefore follow Grenager and Manning (2006) in treating linkings as first-class objects; however, we differ from their work in two important respects. First, we use semantic clustering of head words of arguments in an approach that resembles topic modeling, rather than directly modeling the subcategorization of verbs with a distribution over words. Second and most importantly, we do not make any assumptions about the linkings, as do Grenager and Manning (2006). They list a small set of rules from which they derive all linkings possible in their model; in contrast, we are able to learn any linking observed in the data. Therefore, our approach is language-independent. Grenager and Manning (2006) claim that their rules represent "a weak form of Universal Grammar", but their rules lack such common linking operations as the addition of an accusative reflexive for the unaccusative (Romance) or case marking (many languages), and they include a specific (English) preposition. We have no objection to using linguistic knowledge, but we do not feel that we have the empirical basis as of now to provide a set of Universal Grammar rules relevant for our task.

A complete syntax-semantics lexicon describes how lexemes syntactically realize their semantic arguments, and provides selectional preferences on these dependents. Though rich lexical resources exist (such as the PropBank rolesets, the FrameNet lexicon, or VerbNet, which relates and extends these sources), none of them is complete, not even for English, on which most of the efforts have focused. However, if a complete syntax-semantics lexicon did exist, it would be an extremely useful resource: the task of shallow semantic parsing (semantic argument detection and semantic role labeling) could be reduced to determining the best analysis according to this lexicon. In fact, the learning model we present in this paper is itself a semantic role labeling model, since we can simply apply it to the data we want to label semantically.

This paper is a step towards the unsupervised induction of a complete syntax-semantics lexicon. We present a unified procedure for associating verbs with linkings and for associating the discovered semantic roles with selectional preferences. As input, we assume a syntactic representation scheme and a parser which can produce syntactic representations of unseen sentences in the chosen scheme reasonably well, as well as unlabeled text. We do not assume a specific theory of lexical semantics, nor a specific set of semantic roles. We induce a set of linkings, which are mappings from semantic role symbols to syntactic functions. We also induce a lexicon, which associates a verb lemma with a distribution over the linkings, and which associates the sematic role symbols with verb-specific selectional preferences (which are distributions over distributions of words). We evaluate on the task of semantic role labeling using PropBank (Palmer et al., 2005) as a gold standard.

We focus on semantic arguments, as they are defined specifically for each verb and thus have verb-specific mappings to syntactic arguments, which may further be subject to diathesis alternations. In contrast, semantic adjuncts (modifiers) apply (in principle) to all verbs, and do not participate in diathesis alternations. For this reason, the PropBank lexicon includes arguments but not adjuncts in its framesets. The method we present in this paper is designed to find verb-specific arguments, and we therefore take the results on semantic arguments (Arg$n$) as our primary result. On these, we achieve a 20% F-measure error reduction over a high syntactic baseline (which maps each syntactic relation to a single semantic argument).

2 Related Work

As mentioned above, our approach is most similar to that of Grenager and Manning (2006). However, since their model uses hand-crafted rules, they are able to predict and evaluate against actual PropBank role labels, whereas our approach has to be evaluated in terms of clustering quality.

The problem of unsupervised semantic role labeling has recently attracted some attention (Lang and Lapata, 2011a; Lang and Lapata, 2011b; Titov and Klementiev, 2012). While the present paper shares the general aim of inducing semantic role clusters in an unsupervised way, it differs in treating syntax-semantics linkings explicitly and modeling predicate-specific distributions over them.

Abend et al. (2009) address the problem of unsupervised argument recognition, which we do not address in the present paper. For the purpose of building a complete unsupervised semantic parser, a method such as theirs would be complementary to our work.

3 Model

In this section, we decribe a model that generates arguments for a given predicate instance. Specifically, this generative model describes the probability of a given set of argument head words and associated syntactic functions in terms of underlying semantic roles, which are modelled as latent variables. The semantic role labeling task is therefore framed as the induction of these latent variables from the observed data, which we assume to be preprocessed by a syntactic parser.

The basic idea of our approach is to explicitly model *linkings* between the syntactic realizations and the underlying semantic roles of the arguments in a predicate-argument structure. Since our model of argument classification is completely unsupervised, we cannot assign familiar semantic role labels like *Agent* or *Instrument*, but rather aim at inducing *role clusters*, i.e., clusters of argument instances that share a semantic role. For example, each of the three

instances of *response rate* in (1) should be assigned to the same cluster. We assume a fixed maximum number R of semantic roles per predicate and formulate argument classification as the task of assigning each argument in a predicate-argument structure to one of the numbered roles $1, \ldots, R$. Such an assignment can therefore be represented by an R-tuple, where each role position is either filled by one of the arguments or empty (denoted as ϵ). We represent each argument by its *head word* and its *syntactic function*, i.e., the path of syntactic dependency relations leading to it from the predicate. In our example (1a), a possible assignment of arguments to semantic roles could therefore be represented by a head word tuple $(\text{we}, \text{rate}, \epsilon, \text{SHK})$ and a corresponding tuple of syntactic functions $(\text{nsubj}, \text{dobj}, \epsilon, \text{prep\_with})$, where for the sake of the example we have chosen $R = 4$ and the third semantic role slot is empty. Note that this ordered R-tuple thus represents a semantic labeling of the unordered set of arguments, which our model takes as input. While in the case of a single predicate-argument structure the assignment of arguments to arbitrary semantic role numbers does not provide additional information, its value lies in the consistent assignment of arguments to specific roles *across instances of the same predicate*. For example, to be consistent with the assignment above, (1b) would have to be represented by $(\epsilon, \text{rate}, \epsilon, \text{SHK})$ and $(\epsilon, \text{dobj}, \epsilon, \text{nsubj})$.

To formulate a generative model of argument tuples, we separately consider the tuple of argument head words and the tuple of syntactic functions. The following two subsections will address each of these in turn.

3.1 Selectional Preferences

The probability of an argument in a certain semantic role depends strongly on the *selectional preferences* of the predicate with respect to this role. In the context of our model, we therefore need to describe the probability $P(w_r|p, r)$ of an argument head word w_r depending on the predicate p and the role r. Instead of directly modeling predicate- and role-specific distributions over head words, however, we model selectional preferences as distributions $\chi_{p,r}(c)$ over *semantic word classes* $c = 1, \ldots, C$ (with C being a fixed model parameter), each of which is in turn associated with a distribution $\psi_c(w_r)$ over the vocabulary. They are thus similar to topics in semantic topic models. An advantage of this approach is that semantic word classes can be shared among different predicates, which facilitates their inference. Technically, the introduction of semantic word classes can be seen as a factorization of the probability of the argument head $P(w_r|p, r) = \sum_{c=1}^{C} \chi_{p,r}(c)\psi_c(w_r)$.

3.2 Linkings

Another important factor for the assignment of arguments to semantic roles are their syntactic functions. While in the preceding subsection we considered selectional preferences for each semantic role separately (assuming their independence), the interdependence between syntactic functions is crucial and cannot be ignored: The assignment of an argument does not depend solely on its own syntactic function, but on the whole *subcategorization frame* of the predicate-argument structure. We therefore have to model the probability of the whole tuple $y = (y_1, \ldots, y_R)$ of syntactic functions.

We assume that for each predicate there is a relatively small number of ways in which it realizes its arguments syntactically, i.e., in which semantic roles are linked to syntactic functions. These may correspond to alternations like those shown in (1). Instead of directly modeling the predicate-specific probability $P(y|p)$, we consider predicate-specific distributions $\phi_p(l)$ over linkings $l = (x_1, \ldots, x_R)$. Such a linking then gives rise to the tuple $y = (y_1, \ldots, y_R)$ by way of probability distributions $P(y_r|x_r) = \eta_{x_r}(y_r)$. This allows us to keep the number of possible linkings l per predicate relatively small (by setting $\phi_p(l) = 0$ for most l), and generate a wide variety of syntactic function tuples y from them.

3.3 Structure of the Model

Figure 1 presents our linking model. For each predicate-argument structure in the corpus, it contains observable variables for the predicate p and the unordered set s of arguments, and further shows latent variables for the linking l and (for each role r) the semantic word class c, the head word w, and the syntactic function y.

The distributions $\chi_{p,r}(c)$ and $\psi_c(w)$ are drawn from Dirichlet priors with symmetric parameters α and β, respectively. In the case of the linking dis-

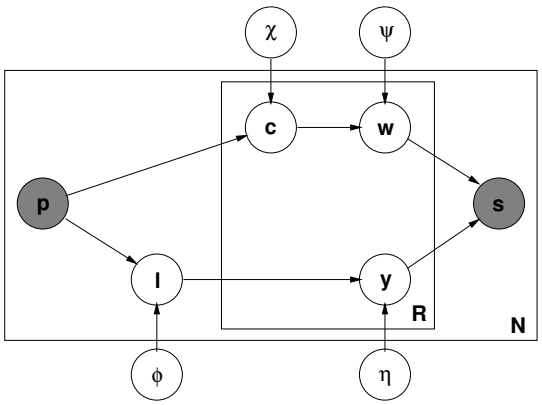

Figure 1: Representation of our linking model as a Bayesian network. The nodes p and s are observed for each of the N predicate-argument structures in the corpus. The latent variables c, w, l, and y are inferred from the data along with their distributions χ, ψ, ϕ, and η.

tribution $\phi_p(l)$, we are faced with an exponentially large space of possible linkings (considering a set G of syntactic functions, there are $(|G|+1)^R$ possible linkings). This is both computationally problematic and counter-intuitive. We therefore maintain a global list L of permissible linkings and enforce $\phi_p(l) = 0$ for all $l \notin L$. On the set L we then draw $\phi_p(l)$ from a Dirichlet prior with symmetric parameter γ. In Section 3.5, we will describe how the linking list L is iteratively induced from the data.

We introduced the distribution η_x to allow for incidental changes when generating the tuple of syntactic functions out of the linking. If this process were allowed to arbitrarily change any syntactic function in the linking, the linkings would be too unconstrained and not reflect the syntactic functions in the corpus. We therefore parameterize η_x in such a way that the only allowed modifications are the addition or removal of syntactic functions from the linking, but no change from one syntactic function to another. We attain this by parameterizing η_x as follows:

$$\eta_x(y) = \begin{cases} \eta^\epsilon & \text{if } x = y = \epsilon \\ \frac{1-\eta^\epsilon}{|G|} & \text{if } x = \epsilon \text{ and } y \in G \\ 1-\eta^x & \text{if } x \in G \text{ and } y = \epsilon \\ \eta^x & \text{if } x = y \in G \\ 0 & \text{else} \end{cases}$$

Here, G again denotes the set of all syntactic functions. The parameter η^ϵ is drawn from a uniform prior on the interval $[0.0, 1.0]$ and the $|G|$ parameters η^x for $x \in G$ have uniform priors on $[0.5, 1.0]$. This has the effect that no syntactic function can change into another, that a syntactic function is never more probable to disappear than to stay, and that all syntactic functions are added with the same probability. This last property will be important for the iterative refinement process described in Section 3.5.

3.4 Training

In this subsection, we describe how we train the model described so far, assuming that we are given a fixed linking list L. The following subsection will address the problem of infering this list. In Section 3.6, we will then describe how we apply the trained model to infer semantic role assignments for given predicate-argument structures.

To train the linking model, we apply a Gibbs sampling procedure to the latent variables shown in Figure 1. In each sampling iteration, we first sample the values of the latent variables of each predicate-argument structure based on the current distributions, and then the latent distributions based on counts obtained over the corpus. For each predicate-argument structure, we begin with a blocked sampling step, simultaneously drawing values for w and y, while summing out c. This gives us

$$P(w, y|p, l, s) \propto \prod_{r=1}^{R} \eta_{x_r}(y_r) \sum_{c=1}^{C} \chi_{p,r}(c)\psi_c(w_r)$$

where we have omitted the factor $P(s|w, y)$, which is uniform as long as we assume that w and y indeed represent permutations of the argument set s. To sample efficiently from this distribution, we precompute the inner sum (as a tensor contraction or, equivalently, R matrix multiplications). We then enumerate all permutations of the argument set and compute their probabilities, defaulting to an approximative beam search procedure in cases where the space of permutations is too large.

Next, the linking l is sampled according to

$$P(l|p, y) \propto P(l|p)P(y|l) = \phi_p(l) \prod_{r=1}^{R} \eta_{x_r}(y_r)$$

Since the space L of possible linkings is small, completely enumerating the values of this distribution is

not a problem.

After sampling the latent variables w, y, and l for each corpus instance, we go on to apply Gibbs sampling to the latent distributions. For example, for ϕ_p we obtain

$$P(\phi_p|p^1, l^1, \ldots, p^N, l^N) \propto P(\phi_p) \prod_{i=1}^{N} P(l^i|p^i)$$

$$\propto \mathrm{Dir}(\gamma)(\phi_p) \cdot \prod_{l \in L} [\phi_p(l)]^{n_p(l)} = \mathrm{Dir}(\vec{n}_p + \gamma)(\phi_p)$$

Here $n_p(l)$ is the number of corpus instances with predicate p and latent linking l, and $\vec{n}_p$ is the vector of these counts for a fixed p, indexed by l. Hence, ϕ_p is drawn from the Dirichlet distribution parameterized by this vector, smoothed in each component by γ.

In the same way, the sampling distributions for $\chi_{p,r}$ and ψ_c are determined as $\mathrm{Dir}(\vec{n}_{p,r} + \alpha)$ and $\mathrm{Dir}(\vec{n}_c + \beta)$, where each $\vec{n}_{p,r}$ is a vector of counts[1] indexed by word classes c and each $\vec{n}_c$ is a vector of counts indexed by head words w_r. Similarly, we draw the parameter η^ϵ in the parameterization of η_x from $\mathrm{Beta}\left(n(\epsilon, \epsilon) + 1, \sum_{x \in G} n(\epsilon, x) + 1\right)$ and approximate η^x by drawing η^x from $\mathrm{Beta}\left(n(x, x) + 1, n(x, \epsilon) + 1\right)$ and redrawing it uniformly from $[0.5, 1.0]$, if it is smaller than 0.5. In this context, $n(x, y)$ refers to the number of times the syntactic relation x is turned into y, counted over all corpus instances and semantic roles.

To test for convergence of the sampling process, we monitor the log-likelihood of the data. For each predicate-argument structure with predicate p^i and argument set s^i, we have

$$P(p^i, s^i) \propto \sum_l P(l|p^i) P(s^i|l) \approx P(s^i|l^i)$$

$$= \sum_{w,y} P(w, y, s^i|l^i) = \sum_{w,y \Rightarrow s^i} P(w, y|l^i) =: L_i$$

The approximation is rather crude (replacing an expected value by a single sample from $P(l|p^i)$), but we expect the errors to mostly cancel out over the instances of the corpus. The last sum ranges over all pairs (w, y) that represent permutations of the argument set s, and this can be computed as a by-product of the sampling process of w and y. We then compute $L := \log \prod_{i=1}^{N} L_i = \sum_{i=1}^{N} \log L_i$, and terminate the sampling process if L does not increase by more than 0.1% over 5 iterations.

[1] Since we do not sample c, we use pseudo-counts based on $P(c_r|p, r, w_r)$ for each instance.

3.5 Iterative Refinement of Possible Linkings

In Section 3.3, we have addressed the problem of the exponentially large space of possible linkings by introducing a subset $L \subset G^R$ from which linkings may be drawn. We now need to clarify how this subset is determined. In contrast to Grenager and Manning (2006), we do not want to use any linguistic intuitions or manual rules to specify this subset, but rather automatically infer it from the data, so that the model stays agnostic to the language and paradigm of semantic roles. We therefore adopt a strategy of *iterative refinement*.

We start with a very small set that only contains the trivial linking $(\epsilon, \ldots, \epsilon)$ and one linking for each of the R most frequent syntactic functions, placing the most frequent one in the first slot, the second one in the second slot etc. We then run Gibbs sampling. When it has converged in terms of log-likelihood, we add some new linkings to L. These new linkings are inferred by inspecting the action of the step from l to y in the generative model. Here, a syntactic function may be added to or deleted from a linking. If a particular syntactic function is frequently added to some linking, then a corresponding linking, i.e., one featuring this syntactic function and thus not requiring such a modification, seems to be missing from the set L. We therefore count for each linking l how often it is either reduced by the deletion of any syntactic function or expanded by the addition of a syntactic function. We then rank these modifications in descending order and for each of them determine the semantic role slot in which the modification (deletion or addition) occured most frequently. By applying the modification to this slot, each of the linkings gives rise to a new one. We add the first a of those, skipping new linkings if they are duplicates of those we already have in the linking set. We iterate this procedure, alternating between Gibbs sampling to convergence and the addition of a new linkings.

3.6 Inference

To predict semantic roles for a given predicate and argument set, we maximize $P(l, w, y|p, s)$. If the

space of permutations is too large for exhaustive enumeration, we apply a similar beam search procedure as the one employed in training to approximately maximize $P(w, y|p, s, l)$ for each value of l. For efficiency, we do not marginalize over l. This has the potential of reducing prediction quality, as we do not predict the most likely role assignment, but rather the most likely combination of role assignment and latent linking.

In all experiments we averaged over 10 consecutive samples of the latent distributions, at the end of the sampling process (i.e., when convergence has been reached).

4 Experimental Setup

We train and evaluate our linking model on the data set produced for the CoNLL-08 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies (Surdeanu et al., 2008), which is based on the PropBank corpus (Palmer et al., 2005). This data set includes part-of-speech tags, lemmatized tokens, and syntactic dependencies, which have been converted from the manual syntactic annotation of the underlying Penn Treebank (Marcus et al., 1993).

4.1 Data Set

As input to our model, we decided not to use the syntactic representation in the CoNLL-08 data set, but instead to rely on Stanford Dependencies (de Marneffe et al., 2006), which seem to facilitate semantic analysis. We thus used the Stanford Parser[2] to convert the underlying phrase structure trees of the Penn Tree Bank into Stanford Dependencies. In the resulting dependency analyses, the syntactic head word of a semantic role may differ from the syntactic head according to the provided syntax. We therefore mapped the semantic role annotation onto the Stanford Dependency trees by identifying the tree node that covers the same set of tokens as the one marked in the CoNLL-08 data set.

The focus of the present work is on the linking behavior and classification of semantic arguments and not their identification. The latter is a substantially different task, and likely to be best addressed by other approaches, such as that of (Abend et al., 2009). We therefore use gold standard information of the CoNLL-08 data set for identifying argument sets as input to our model. The task of our model is then to *classify* these arguments into semantic roles.

We train our model on a corpus consisting of the training and the test part of the CoNLL-08 data set, which is permissible since as a unsupervised system our model does not make any use of the annotated argument labels for training. We test the model performance against the gold argument classification on the test part. For development purposes (both designing the model and tuning the parameters as described in Section 4.4), we train on the training and development part and test on the development part.

4.2 Evaluation Measures

As explained above, our model does not predict specific role labels, such as those annotated in PropBank, but rather aims at clustering like argument instances together. Since the (numbered) labels of these clusters are arbitrary, we cannot evaluate the predictions of our model against the PropBank gold annotation directly. We follow Lang and Lapata (2011b) in measuring the quality of our clustering in terms of cluster purity and collocation instead.

Cluster purity is a measure of the degree to which the predicted clusters meet the goal of containing only instances with the same gold standard class label. Given predicted clusters $C_1, \ldots, C_{n_C}$ and gold clusters $G_1, \ldots, G_{n_G}$ over a set of n argument instances, it is defined as

$$\text{Pu} = \frac{1}{n} \sum_{i=1}^{n_C} \max_{j=1,\ldots,n_G} |C_i \cap G_j|$$

Similarly, cluster collocation measures how well the clustering meets the goal of clustering all gold instances with the same label into a single predicted cluster, formally:

$$\text{Co} = \frac{1}{n} \sum_{j=1}^{n_G} \max_{i=1,\ldots,n_C} |C_i \cap G_j|$$

We determine purity and collocation separately for each predicate type and then compute their micro-average, i.e., weighting each score by the number of argument instances of this precidate. Just as precision and recall, purity and collocation stand in tradeoff. In the next section, we therefore report their F_1 score, i.e., their harmonic mean $\frac{2 \cdot Pu \cdot Co}{Pu + Co}$.

[2] version 1.6.8, available at `http://nlp.stanford.edu/software/lex-parser.shtml`

4.3 Syntactic Baseline

We compare the performance of our model with a simple syntactic baseline that assumes that semantic roles are identical with syntactic functions. We follow Lang and Lapata (2011b) in clustering argument instances of each predicate by their syntactic functions. We do not restrict the number of clusters per predicate. In contrast, Lang and Lapata (2011b) restrict the number of clusters to 21, which is the number of clusters their system generates. We found that this reduces the baseline by 0.1% F_1-score (Arg$n$ on the development set, c.f. Table 1). If we reduce the number of clusters in the baseline to the number of clusters in our system (7), the baseline is reduced by another 0.8% F_1-score. These lower baselines are due to lower purity values. In general, we find that a smaller number of clusters results in lower F_1 measure for the baseline; the reported baseline therefore is the strictest possible.

4.4 Parameters and Tuning

For all experiments, we fixed the number of semantic roles at $R = 7$. This is the maximum size of the argument set over all instances of the data set and thus the lower limit for R. If R was set to a higher value, the model would be able to account for the possibility of a larger number of roles, out of which never more than 7 are expressed simultaneously. We leave such investigation to future work. We set the symmetric parameters for the Dirichlet distributions to $\alpha = 1.0$, $\beta = 0.1$, and $\gamma = 1.0$. This corresponds to uninformative uniform priors for $\chi_{p,r}$ and ϕ_p, and a prior encouraging a sparse lexical distribution ψ_c, similar as in topic models such as LDA (Blei et al., 2003).

The number C of word classes, the number a of additional linkings in each refinement of the linking set L, and the number k of refinement steps were tuned on the development set. We first fixed $a = 10$ and trained models for $C = 10, 20, \ldots, 100$, performing 50 refinement steps. The best F_1 score was obtained with $C = 10$ after $k = 20$ refinements (i.e., with 200 linkings). Next, we fixed these two parameters and trained models for $a = 5, 10, 15, 20, 25$. Here, we confirmed an optimal value of $a = 10$.

5 Results

In this section, we give quantitative results, comparing our system to the syntactic baseline in terms of cluster purity and collocation, and a qualitative discussion of some phenomena observed in the performance of the model.

5.1 Quantitative Results

Table 1 shows the results of applying our models to the CoNLL-08 test with the parameter values tuned in Section 4.4. For comparison, we also show results on the development set. The table is divided into three parts, one only considering semantic arguments (Arg$n$), one considering adjuncts (ArgM), and one aggregating results over both kinds of PropBank roles (Arg*). It can be seen that our model consistently outperforms the syntactic baseline in terms of collocation (by 10% on Arg$n$, 3% on ArgM, and 8.2% overall). In terms of purity, however, it falls short of the baseline. As mentioned above, there is a trade-off between purity and collocation. Compared to our model, which we run with a total of 7 semantic role slots, the baseline predicts a large number of small argument clusters for each predicate, whereas our model tends to group arguments together based on selectional preferences.

In terms of F_1 score, our model outperforms the baseline by 3.6% on Arg$n$, which translates into a relative error reduction by 20%. On adjuncts, on the other hand, our model falls short of the baseline by almost 10% F_1 score. This indicates that our approach based on explicit representations of linkings is most suited to the classification of arguments rather than adjuncts, which do not participate in diathesis alternations and do therefore not profit as much from our explicit induction of linkings.

5.2 Qualitative Observations

Among the verbs with at least 10 test instances, *include* shows the largest gain in F_1 score over the baseline. In the test corpus, we find an interesting pair of sentences for this predicate:

(2) a. *Mr. Herscu proceeded to launch an ambitious, but ill-fated, $1 billion acquisition binge that included Bonwit Teller and B. Altman & Co. [...]*

	Arg$n$			ArgM			Arg*		
Test Set	Pu	Co	F_1	Pu	Co	F_1	Pu	Co	F_1
Syntactic Baseline	**90.6**	75.4	82.3	**87.0**	73.3	**79.6**	**88.0**	74.9	**80.9**
Linking Model	86.4	**85.4**	**85.9**	64.4	**76.3**	69.8	74.5	**83.1**	78.6
Development Set	Pu	Co	F_1	Pu	Co	F_1	Pu	Co	F_1
Syntactic Baseline	**91.5**	73.9	81.8	**88.7**	78.6	**83.3**	**89.2**	75.1	**81.5**
Linking Model	85.6	**84.4**	**85.0**	67.7	**79.9**	73.3	75.2	**83.2**	79.0

Table 1: Purity (Pu), collocation (Co), and F_1 scores of our model and the syntactic baseline in percent. Performance on arguments (Arg$n$), adjuncts (ArgM), and overall results (Arg*) are shown separately.

b. *Not included in the bid are Bonwit Teller or B. Altman & Co. [...]*

The first of these two sentences is generated from the linking $(\text{nsubj}, \text{dobj}, \epsilon, \epsilon, \epsilon, \epsilon, \text{-rcmod})$, which does not need to be modified in any way to account for the subject *that* (coreferent with the head of the predicate in the modifying relative clause, *binge*) and the direct object *Teller* (head of the phrase *Bonwit Teller and B. Altman & Co.*). These are assigned to the first and second role slots, respectively. The second sentence, on the other hand, is generated out of the linking $(\text{prep\_in}, \text{nsubjpass}, \epsilon, \epsilon, \epsilon, \epsilon, \epsilon)$. Here, the passive subject *Teller* is assigned to the second role slot (which we may interpret as the *Includee*), while the first semantic role (the *Includer*) is labeled on *bid*, which is realized in a prepositional phrase headed by the preposition *in*. Note that this alternation is not the general passive alternation though, which would have led to *Teller is not included **by** the bid*. Instead, the model learned a specific alternation pattern for the predicate *include*.

But even where a specific linking has not been learned, the model can often still infer a correct labeling by virtue of its selectional preference component. In our corpus, the predicate *give* occurs mostly with a direct and an indirect object as in *CNN recently gave most employees raises of as much as 15%*. The model therefore learns a linking $(\text{nsubj}, \text{dobj}, \epsilon, \epsilon, \epsilon, \epsilon, \text{iobj})$, but fails to learn that the *Beneficient* role can also be expressed with the preposition *to* as in

(3) *[...] only 25% give $2,500 or more to charity each year.*

However, when applying our model to this sentence, it nonetheless assigns *charity* to the last role slot (the same one previously occupied by the indirect object). This is due to the fact that *charity* is a good fit for the selectional preference of this role slot of the predicate *give*.

6 Conclusions

We have presented a novel generative model of predicate-argument structures that incorporates selectional preferences of argument heads and explicitly describes linkings between semantic roles and syntactic functions. The model iteratively induces a lexicon of possible linkings from unlabeled data. The trained model can be used to cluster given argument instances according to their semantic roles, outperforming a competitive syntactic baseline.

The approach is independent of any particular language or paradigm of semantic roles. However, in its present form the model assumes that each predicate has its own set of semantic roles. In formalisms such as Frame Semantics (Baker et al., 1998), semantic roles generalize across semantically similar predicates belonging to the same *frame*. A natural extension of our approach would therefore consist in modeling predicate groups that share semantic roles and selectional preferences.

Acknowledgments. This work was supported by the Intelligence Advanced Research Projects Activity (IARPA) via Department of Interior National Business Center (DoI/NBC) contract number D11PC20153. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. Disclaimer: The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of IARPA, DoI/NBC, or the U.S. Government.

References

Omri Abend, Roi Reichart, and Ari Rappoport. 2009. Unsupervised argument identification for semantic role labeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 28–36, Singapore.

Collin F. Baker, J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *36th Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics (COLING-ACL'98)*, pages 86–90, Montréal.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Luigi Burzio. 1986. *Italian Syntax: A Government-Binding Approach*. Reidel, Dordrecht.

Marie-Catherine de Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC 2006*.

Trond Grenager and Christopher D. Manning. 2006. Unsupervised discovery of a statistical verb lexicon. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 1–8, Sydney, Australia.

Joel Lang and Mirella Lapata. 2011a. Unsupervised semantic role induction via split-merge clustering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1117–1126, Portland, Oregon, USA.

Joel Lang and Mirella Lapata. 2011b. Unsupervised semantic role induction with graph partitioning. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1320–1331, Edinburgh, Scotland, UK.

Maria Lapata and Chris Brew. 1999. Using subcategorization to resolve verb class ambiguity. In *In Proceedings of Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, pages 266—-274, College Park, MD.

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

Mitchell M. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19.2:313–330, June.

Paola Merlo and Suzanne Stevenson. 2001. Automatic verb classification based on statistical distributions of argument structure. *Computational Linguistics*, 27(3).

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The conll 2008 shared task on joint parsing of syntactic and semantic dependencies. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 159–177, Manchester, England.

Ivan Titov and Alexandre Klementiev. 2012. A bayesian approach to unsupervised semantic role induction. In *Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics*, Avignon, France, April.