# Unsupervised Induction of Semantic Roles

**Joel Lang** and **Mirella Lapata**
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB, UK
J.Lang-3@sms.ed.ac.uk, mlap@inf.ed.ac.uk

## Abstract

Datasets annotated with semantic roles are an important prerequisite to developing high-performance role labeling systems. Unfortunately, the reliance on manual annotations, which are both difficult and highly expensive to produce, presents a major obstacle to the widespread application of these systems across different languages and text genres. In this paper we describe a method for inducing the semantic roles of verbal arguments directly from unannotated text. We formulate the role induction problem as one of detecting alternations and finding a canonical syntactic form for them. Both steps are implemented in a novel probabilistic model, a latent-variable variant of the logistic classifier. Our method increases the purity of the induced role clusters by a wide margin over a strong baseline.

## 1 Introduction

Semantic role labeling (SRL, Gildea and Jurafsky 2002) is the task of automatically classifying the arguments of a predicate with roles such as *Agent*, *Patient* or *Location*. These labels capture aspects of the semantics of the relationship between the predicate and the argument while abstracting over surface syntactic configurations. SRL has received much attention in recent years (Surdeanu et al., 2008; Màrquez et al., 2008), partly because of its potential to improve applications that require broad coverage semantic processing. Examples include information extraction (Surdeanu et al., 2003), question answering (Shen and Lapata, 2007), summarization (Melli et al., 2005), and machine translation (Wu and Fung, 2009).

Given sentences (1-a) and (1-b) as input, an SRL system would have to identify the verb predicate (shown in boldface), its arguments (*Michael* and *sandwich*) and label them with semantic roles (*Agent* and *Patient*).

(1) a. [Michael]$_{Agent}$ **eats** [a sandwich]$_{Patient}$.
b. [A sandwich]$_{Patient}$ **is eaten** [by Michael]$_{Agent}$.

Here, sentence (1-b) is an alternation of (1-a). The verbal arguments bear the same semantic role, even though they appear in different syntactic positions: *sandwich* is the object of *eat* in sentence (1-a) and its subject in (1-b) but it is in both instances assigned the role *Patient*. The example illustrates the passive alternation. The latter is merely one type of alternation, many others exist (Levin, 1993), and their computational treatment is one of the main challenges faced by semantic role labelers.

Most SRL systems to date conceptualize semantic role labeling as a supervised learning problem and rely on role-annotated data for model training. PropBank (Palmer et al., 2005) has been widely used for the development of semantic role labelers as well as FrameNet (Fillmore et al., 2003). Under the PropBank annotation framework (which we will assume throughout this paper) each predicate is associated with a set of core roles (named $A0$, $A1$, $A2$, and so on) whose interpretations are specific to that predicate[1] and a set of adjunct roles (e.g., *Location* or *Time*) whose interpretation is common across predicates. In addition to large amounts of role-annotated data, SRL systems often make use of a parser to obtain syntactic analyses which subsequently serve as input to a pipeline of components concerned with

[1] More precisely, $A0$ and $A1$ have a common interpretation across predicates as *proto-agent* and *proto-patient* (Dowty, 1991).

*Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the ACL*, pages 939–947,
Los Angeles, California, June 2010. ©2010 Association for Computational Linguistics

identifying predicates and their arguments (argument identification) and labeling them with semantic roles (argument classification).

Supervised SRL methods deliver reasonably good performance (a system will recall around 81% of the arguments correctly and 95% of those will be assigned a correct semantic role; see Màrquez et al. 2008 for details). Unfortunately, the reliance on labeled training data, which is both difficult and highly expensive to produce, presents a major obstacle to the widespread application of semantic role labeling across different languages and text genres. And although corpora with semantic role annotations exist nowadays in other languages (e.g., German, Spanish, Catalan, Chinese, Korean), they tend to be smaller than their English equivalents and of limited value for modeling purposes. Moreover, the performance of supervised systems degrades considerably (by 10%) on out-of-domain data even within English, a language for which two major annotated corpora are available. Interestingly, Pradhan et al. (2008) find that the main reason for this are errors in the assignment of semantic roles, rather than the identification of argument boundaries. Therefore, a mechanism for inducing the semantic roles observed in the data without additional manual effort would enhance the robustness of existing SRL systems and enable their portability to languages for which annotations are unavailable or sparse.

In this paper we describe an unsupervised approach to argument classification or *role induction*[2] that does not make use of role-annotated data. Role induction can be naturally formalized as a clustering problem where argument instances are assigned to clusters. Ideally, each cluster should contain arguments corresponding to a specific semantic role and each role should correspond to exactly one cluster. A key insight in our approach is that many predicates are associated with a standard linking. A linking is a deterministic mapping from semantic roles onto syntactic functions such as *subject*, or *object*. Most predicates will exhibit a standard linking, i.e., they will be predominantly used with a specific mapping. Alternations occur when a different linking is used. In sentence (1-a) the predicate *eat* is used with its standard linking (the *Agent* role is mapped onto the *subject* function and the *Patient* onto the *object*), whereas in sentence (1-b) *eat* is used with its passive-linking (the *Patient* is mapped onto *subject* and the *Agent* appears as a prepositional phrase). When faced with such alternations, we will attempt to determine for each argument the syntactic function it would have had, had the standard linking been used. We will refer to this function as the arguments' *canonical function*, and use the term *canonicalization* to describe the process of inferring these canonical functions in the case of alternations. So, in sentence (1-b) the canonical functions of the arguments *by Michael* and *sandwich* are *subject* and *object*, respectively.

Since linkings are injective, i.e., no two semantic roles are mapped onto the same syntactic function, the canonical function of an argument uniquely references a specific semantic role. We define a probabilistic model for detecting non-standard linkings and for canonicalization. The model specifies a distribution $p(F)$ over the possible canonical functions $F$ of an argument. We present an extension of the logistic classifier with the addition of latent variables which crucially allow to learn generalizations over varying syntactic configurations. Rather than using manually labeled data, we train our model on observed syntactic functions which can be obtained automatically from a parser. These training instances are admittedly noisy but readily available and as we show experimentally a useful data source for inducing semantic roles. Application of the model to a benchmark dataset yields improvements over a strong baseline.

## 2 Related Work

Much previous work on SRL relies on supervised learning methods for both argument identification and argument classification (see Màrquez et al. 2008 for an overview). Most systems use manually annotated resources to train separate classifiers for different SRL subtasks (e.g., Surdeanu et al. 2008). A few approaches adopt semi-supervised learning methods. The idea here is to to alleviate the data requirements for semantic role labeling by extending existing resources through the use of unlabeled data. Swier and Stevenson (2004) induce role labels with a bootstrapping scheme in which the set of labeled instances is iteratively expanded using a classifier trained on previously labeled instances. Padó and Lapata (2009) project role-semantic annotations from an annotated corpus in one language onto an unannotated corpus in another language. And Fürstenau and Lapata (2009) propose a method

[2] We use the term *role induction* rather than *argument classification* for the unsupervised setting.

in which annotations are projected from a source corpus onto a target corpus, however within the same language.

Unsupervised approaches to SRL have been few and far between. Early work on lexicon acquisition focuses on identifying verbal alternations rather than their linkings. This is often done in conjunction with hand-crafted resources such as a taxonomy of possible alternations (McCarthy and Korhonen, 1998) or WordNet (McCarthy, 2002). Lapata (1999) proposes a corpus-based method that is less reliant on taxonomic resources, however focuses only on two specific verb alternations. Other work attempts to cluster verbs into semantic classes (e.g., Levin 1993) on the basis of their alternation behavior (Schulte im Walde and Brew, 2002).

More recently, Abend et al. (2009) propose an unsupervised algorithm for argument identification that relies only on part-of-speech annotations, whereas Grenager and Manning (2006) focus on role induction which they formalize as probabilistic inference in a Bayesian network. Their model defines a joint probability distribution over the particular linking used together with a verb instance and for each verbal argument, its lemma, syntactic function as well as semantic role. Parameters in this model are estimated using the EM algorithm as the training instances include latent variables, namely the semantic roles and linkings. To make inference tractable they limit the set of linkings to a small number and do not distinguish between different types of adjuncts. Our own work also focuses on inducing the semantic roles and the linkings used by each verb. Our approach is conceptually simpler and computationally more tractable. Our model is a straightforward extension of the logistic classifier with latent variables applied to all roles not just coarse ones.

## 3 Problem Formulation

We treat role induction as a clustering problem. The goal is to assign argument instances (i.e., specific arguments, occurring in an input sentence) into clusters such that each cluster contains instances with the same semantic role, and each semantic role is found in exactly one cluster. As we assume PropBank-style roles (Palmer et al., 2005), our model will allocate a separate set of clusters for each predicate and assign the arguments of a specific predicate to one of the clusters associated with it.

| | A0 | A1 | TMP | MNR |
|---|---|---|---|---|
| SBJ | 54514 | 19684 | 15 | 7 |
| OBJ | 3359 | 51730 | 93 | 54 |
| ADV | 162 | 3506 | 976 | 2308 |
| TMP | 5 | 60 | 15167 | 22 |
| PMOD | 2466 | 4860 | 142 | 62 |
| OPRD | 37 | 5554 | 1 | 36 |
| LOC | 17 | 145 | 43 | 157 |
| DIR | 0 | 178 | 15 | 6 |
| MNR | 5 | 48 | 13 | 3312 |
| PRP | 9 | 50 | 11 | 6 |
| LGS | 2168 | 36 | 2 | 2 |
| PRD | 413 | 830 | 31 | 38 |
| NMOD | 422 | 388 | 25 | 59 |
| EXT | 0 | 20 | 2 | 12 |
| DEP | 18 | 150 | 25 | 65 |
| SUB | 3 | 84 | 4 | 2 |
| CONJ | 198 | 331 | 22 | 8 |
| ROOT | 62 | 147 | 84 | 2 |
| | 64517 | 88616 | 16803 | 6404 |

Table 1: Contingency table between syntactic function and semantic role for two core roles *Agent* (A0) and *Patient* (A1) and two adjunct roles, *Time* (TMP) and *Manner* (MNR). Only syntactic functions occurring more than 1000 times are listed. Counts were obtained from the CoNLL 2008 training dataset using gold standard parses (the marginals in the bottom row also include counts of unlisted co-occurrences).

As mentioned earlier (Section 1) a linking is a deterministic mapping from semantic roles onto syntactic functions. Table 1 shows how frequently individual semantic roles map onto certain syntactic functions. The frequencies were obtained from the CoNLL 2008 dataset (see Surdeanu et al. 2008 for details) and constitute an aggregate across predicates. As can be seen, there is a clear tendency for a semantic role to be mapped onto a single syntactic function. This is true across predicates and even more so for individual predicates. For example, A0 is commonly mapped onto subject (SBJ), whereas A1 is often realized as object (OBJ). There are two reasons for this. Firstly, a predicate is often associated with a standard linking which is most frequently used. Secondly, the alternate linkings of a given predicate often differ from the standard linking only with respect to a few roles. Importantly, we do not assume that a single standard linking is valid for all predicates. Rather, each predicate has its own standard linking. For example, in the standard linking for the predicate *fall*, A1 is mapped onto subject position, whereas in the standarad linking for *eat*, A1 is mapped onto object position.

When an argument is attested with a non-standard linking, we wish to determine the syntactic function it would have had if the standard linking had been used. This *canonical function* of the argument uniquely references a specific semantic role, i.e., the semantic role that is mapped onto the function under the standard linking. We can now specify an indirect method for partitioning argument instances into clusters:

1. Detect arguments that are linked in a non-standard way (detection).
2. Determine the canonical function of these arguments (canonicalization). For arguments with standard linkings, their syntactic function corresponds directly to the canonical function.
3. Assign arguments to a cluster according to their canonical function.

We distinguish between detecting non-standard linkings and canonicalization because in principle two separate models could be used. In our probabilistic formulation, both detection and canonicalization rely on an estimate of the probability distribution $p(F)$ over the canonical function $F$ of an argument. When the most likely canonical function differs from the observed syntactic function this indicates that a non-standard linking has been used (detection). This most likely canonical function can be taken as the canonical function of the argument (canonicalization).

Arguments are assigned to clusters based on their inferred canonical function. Since we assume predicate-specific roles, we induce a separate cluster for each predicate. Given $K$ clusters, we use the following scheme for determining the mapping from functions to clusters:

1. Order the functions by occurrence frequency.
2. For each of the $K-1$ most frequent functions allocate a separate cluster.
3. Assign all remaining functions to the $K$-th cluster.

## 4 Model

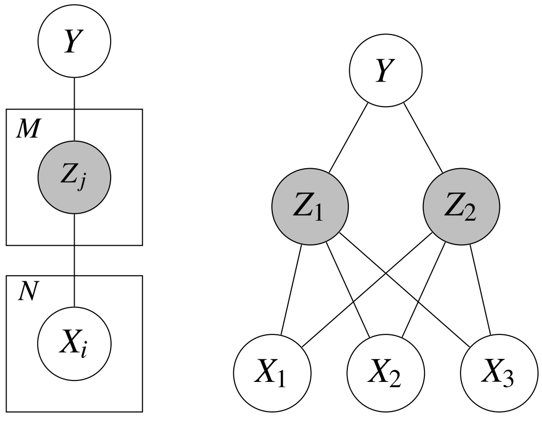
The detection of non-standard linkings and canonicalization both rely on a probabilistic model $p(F)$ which specifies the distribution over the canonical functions $F$ of an argument. As is the case with most SRL approaches, we assume to be given a syntactic parse of the sentence from which we can extract labeled dependencies, corresponding to the syntactic functions of arguments. To train the model we exploit the fact that most observed syntactic functions will correspond to canonical functions. This enables us to use the parser's output for training even though it does not contain semantic role annotations.

Critically, the features used to determine the canonical function must be restricted so that they give no cues about possible alternations. If they would, the model could learn to predict alternations, and therefore produce output closer to the observed syntactic rather than canonical function of an argument. To avoid this pitfall we only use features at or below the node representing the argument head in the parse tree apart from the predicate lemma (see Section 5 for details).

Given these local argument features, a simple solution would be to use a standard classifier such as the logistic classifier (Berger et al., 1996) to learn the canonical function of arguments. However, this is problematic, because in our setting the training and application of the classifier happen on the same dataset. The model will over-adapt to the observed targets (i.e., the syntactic functions) and fail to learn appropriate canonical functions. Lexical sparsity is a contributing factor: the parameters associated with sparse lexical features will be unavoidably adjusted so that they are highly indicative of the syntactic function they occur with.

One way to improve generalization is to incorporate a layer of latent variables into the logistic classifier, which mediates between inputs (features defined over parse trees) and target (the canonical function). As a result, inputs and target are no longer directly connected and the information conveyed by the features about the target must be transferred via the latent layer. The model is shown in plate notation in Figure 1a. Here, $X_i$ represents the observed input features, $Y$ the observed target, and $Z_j$ the latent variables. The number of latent variables influences the generalization properties of the model. With too few latent variables too little information will be transferred via the latent variables, whereas with too many latent variables generalization will degrade.

The model defines a probability distribution over the target variable $Y$ and the latent variables $Z$, con-



(a) (b)

Figure 1: The logistic classifier with latent variables (shaded nodes) illustrated as a graphical model using (a) plate notation and (b) in unrolled form for $M=2$ and $N=3$.

ditional on the input variables $X$:

$$p(y,z|x,\theta) = \frac{1}{P(x,\theta)} \exp\left(\sum_k \theta_k \phi_k(x,y,z)\right) \qquad (1)$$

We will assume that the latent variables $Z_i$ are binary. Each of the feature functions $\phi_k$ is associated with a parameter $\theta_k$. The partition function normalizes the distribution:

$$P(x,\theta) = \sum_y \sum_z \exp\left(\sum_k \theta_k \phi_k(x,y,z)\right) \qquad (2)$$

Note that this model is a special case of a conditional random field with latent variables (Sutton and McCallum, 2007) and resembles a neural network with one hidden layer (Bishop, 2006).

Let $(c,d)$ denote a training set of inputs and corresponding targets. The maximum-likelihood parameters can then be obtained by finding the $\theta$ maximizing:

$$\begin{aligned} l(\theta) &= \log p(d|c) \\ &= \textstyle\sum_i \log \sum_z p(d_i,z|c_i) \\ &= \textstyle\sum_i \log \frac{\sum_z \exp(\sum_k \theta_k \phi_k(c_i,d_i,z))}{P(c_i,\theta)} \end{aligned} \qquad (3)$$

And the gradient is given by:

$$\begin{aligned} (\nabla l)_k &= \tfrac{\partial}{\partial\theta_k} l(\theta) \\ &= \textstyle\sum_i \sum_z p(z|d_i,c_i)\phi_k(c_i,d_i,z) \\ &\quad - \textstyle\sum_i \sum_{y,z} p(y,z|c_i)\phi_k(c_i,y,z) \end{aligned} \qquad (4)$$

where the first term is the conditional expected feature count and the second term is the expected feature count.

Thus far, we have written the equations in a generic form for arbitrary conditional random fields with latent variables (Sutton and McCallum, 2007). In our model we have two types of pairwise sufficient statistics: $\beta(x,z) : \mathbb{R} \times \{0,1\} \to \mathbb{R}$, between a single input variable and a single latent variable, and $\gamma(y,z) : \mathcal{Y} \times \{0,1\} \to \mathbb{R}$, between the target and a latent variable. Then, we can more specifically write the gradient component of a parameter associated with a sufficient statistic $\beta(x_j,z_k)$ as:

$$\sum_i \sum_{z_k} p(z_k|d_i,c_i)\beta(c_{i,j},z_k) - \sum_i \sum_{z_k} p(z_k|c_i)\beta(c_{i,j},z_k) \quad (5)$$

And the gradient component of a parameter associated with a sufficient statistic $\gamma(y,z_k)$ is:

$$\sum_i \sum_{z_k} p(z_k|d_i,c_i)\gamma(d_i,z_k) - \sum_i \sum_{y,z_k} p(y,z_k|c_i)\gamma(y,z_k) \quad (6)$$

To obtain maximum-a-posteriori parameter estimates we regularize the equations. Like for the standard logistic classifier this results in an additional term of the target function and each component of the gradient (see Sutton and McCallum 2007). Computing the gradient requires computation of the marginals which can be performed efficiently using belief propagation (Yedidia et al., 2003). Note that due to the fact, that there are no edges between the latent variables, the inference graph is tree structured and therefore inference yields exact results. We use a stochastic gradient optimization method (Bottou, 2004) to optimize the target. Optimization is likely to result in a local maximum, as the likelihood function is not convex due to the latent variables.

## 5 Experimental Design

In this section we discuss the experimental design for assessing the performance of the model described above. We give details on the dataset, features and evaluation measures employed and present the baseline methods used for comparison with our model.

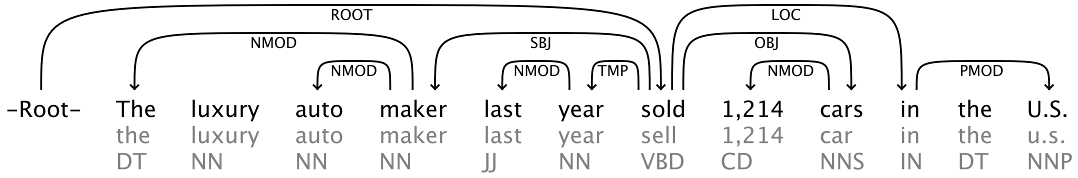

ROOT
NMOD
NMOD
SBJ
NMOD
TMP
LOC
OBJ
NMOD
PMOD
-Root-
The luxury auto maker last year sold 1,214 cars in the U.S.
the luxury auto maker last year sell 1,214 car in the u.s.
DT NN NN NN JJ NN VBD CD NNS IN DT NNP

Figure 2: Dependency graph (simplified) of a sample sentence from the corpus.

**Data** Our experiments were carried out on the CoNLL 2008 (Surdeanu et al., 2008) training dataset which contains both verbal and nominal predicates. However, we focused solely on verbal predicates, following most previous work on semantic role labeling (Màrquez et al., 2008). The CoNLL dataset is taken form the Wall Street Journal portion of the Penn Treebank corpus (Marcus et al., 1993). Role semantic annotations are based on PropBank and have been converted from a constituent-based to a dependency-based representation (see Surdeanu et al. 2008). For each argument of a predicate only the head word is annotated with the corresponding semantic role, rather than the whole constituent. In this paper we are only concerned with role induction, not argument identification. Therefore, we identify the arguments of each predicate by consulting the gold standard.

The CoNLL dataset also supplies an automatic dependency parse of each input sentence obtained from the MaltParser (Nivre et al., 2007). The target and features used in our model are extracted from these parses. Syntactic functions occurring more than $1,000$ times in the gold standard are shown in Table 1 (for more details we refer the interested reader to Surdeanu et al. 2008). Syntactic functions were further modified to include prepositions if specified, resulting in a set of functions with which arguments can be distinguished more precisely. This was often the case with functions such as ADV, TMP, LOC, etc. Also, instead of using the preposition itself as the argument head, we used the actual content word modifying the preposition. We made no attempt to treat split arguments, namely instances where the semantic argument of a predicate has several syntactic heads. These are infrequent in the dataset, they make up for less than 1% of all arguments.

**Model Setup** The specific instantiation of the model used in our experiments has 10 latent variables. With 10 binary latent variables we can encode 1024 different target values, which seems reasonable for our set of syntactic functions which comprises around 350 elements.

Features representing argument instances were extracted from dependency parses like the one shown in Figure 2. We used a relatively small feature set consisting of: the predicate lemma, the argument lemma, the argument part-of-speech, the preposition involved in dependency between predicate and argument (if there is one), the lemma of left-most/right-most child of the argument, the part-of-speech of left-most/right-most child of argument, and a key formed by concatenating all syntactic functions of the argument's children. The features for the argument *maker* in Figure 2 are [*sell, maker, NN, –, the, auto, DT, NN, NMOD+NMOD*]. The target for this instance (and observed syntactic function) is SBJ.

**Evaluation** Evaluating the output of our model is no different from other clustering problems. We can therefore use well-known measures from the clustering literature to assess the quality of our role induction method. We first created a set of gold-standard role labeled argument instances which were obtained from the training partition of the CoNLL 2008 dataset (corresponding to sections 02–21 of PropBank). We used 10 clusters for each predicate and restricted the set of predicates to those attested with more than 20 instances. This rules out simple cases with only few instances relative to the number of clusters, which trivially yield high scores.

We compared the output of our method against the gold-standard using the following common measures. Let $K$ denote the number of clusters, $c_i$ the set of instances in the $i$-th cluster and $g_j$ the set of instances having the $j$-th gold standard semantic role label. Cluster purity (PU) is defined as:

$$PU = \frac{1}{K} \sum_i \max_j |c_i \cap g_j| \qquad (7)$$

We also used cluster accuracy (CA, Equation 8),

|  | PU |  | CA |  | CP |  | CR |  | CF1 |  |
|---|---|---|---|---|---|---|---|---|---|---|
|  | Mic | Mac | Mic | Mac | Mic | Mac | Mic | Mac | Mic | Mac |
| SyntFunc | 73.2 | 75.8 | 82.0 | 80.9 | 67.6 | 65.3 | 55.7 | 50.1 | 61.1 | 56.7 |
| LogLV | 72.5 | 74.0 | 81.1 | 79.4 | 64.3 | 60.6 | 59.7 | 56.3 | 61.9 | 58.4 |
| UpperBndS | 94.7 | 96.1 | 96.9 | 97.0 | 97.4 | 97.6 | 90.4 | 100 | 93.7 | 93.8 |
| UpperBndG | 98.8 | 99.4 | 99.9 | 99.9 | 99.7 | 99.9 | 100 | 100 | 99.8 | 100 |

Table 2: Clustering results using our model (LogLV) against the baseline (SyntFunc) and upper bounds (UpperBndS and UpperBndG).

cluster precision (CP, Equation 9), and cluster recall (CR, Equation 9). Cluster F1 (CF1) is the harmonic mean of precision and recall.

$$CA = \frac{TP+TN}{TP+FP+TN+FN} \tag{8}$$

$$CP = \frac{TP}{TP+FP} \qquad CR = \frac{TP}{TP+FN} \tag{9}$$

Here $TP$ is the number of *pairs of instances* which have the same role and are in the same cluster, $TN$ is the number of pairs of instances which have different roles and are in different clusters, $FP$ is the number of pairs of instances with different roles in the same cluster and $FN$ the number of pairs of instances with the same role in different clusters.

**Baselines and Upper Bound** We compared our model against a baseline that assigns arguments to clusters based on their syntactic function. Here, no attempt is made to correct the roles of arguments in non-standard linkings. We would also like to compare our model against a supervised system. Unfortunately, this is not possible, as we are using the designated CoNLL training set as our test set, and any supervised system trained on this data would achieve unfairly high scores. Therefore, we approximate the performance of a supervised system by clustering instances according to their gold standard role after introducing some noise. Specifically, we randomly selected 5% of the gold standard roles and mapped them to an erroneous role. This roughly corresponds to the clustering which would be induced by a state-of-the-art supervised system with 95% precision. Finally, we also report the results of the true upper bound obtained by clustering the arguments, based on their gold standard semantic role (again using 10 clusters per verb).

## 6 Results

Our results are summarized in Table 2. We report cluster purity, accuracy, precision, recall, and F1 for our latent variable logistic classifier (LogLV) and a baseline that assigns arguments to clusters according to their syntactic function (SyntFunc). The table also includes the gold standard upper bound (UpperBndG) and its supervised proxy (UpperBndS). We report micro- and macro-average scores.[3]

Model scores are quite similar to the baseline, which might suggest that the model is simply replicating the observed data. However, this is not the case: canonical functions differ from observed functions for approximately 27% of the argument instances. If the baseline treated these instances correctly, we would expect it to outperform our model. The fact that it does not, indicates that the baseline error rate is higher precisely on these instances. In other words, the model can help in detecting alternate linkings and thus baseline errors.

We further analyzed our model's ability to detect alternate linkings. Specifically, if we assume a standard linking where model and observation agree and an alternate linking where they disagree, we obtain the following. The number of true positives (correctly detected alternate linkings) is 27,606, the number of false positives (incorrectly marked alternations) is 32,031, the number of true negatives (cases where the model correctly did not detect an alternate linking) is 132,556, and the number of false negatives (alternate linkings that the model should have detected but did not) is 32,516.[4]. The analysis shows that 46% of alternations (baseline errors) are detected.

[3] Micro-averages are computed over instances while macro-averages are computed over verbs.

[4] Note that the true/false positives/negatives here refer to alternate linkings, not to be confused with the true/false positives in equations (8) and (9).

|  | PU |  | CA |  | CP |  | CR |  | CF1 |  |
|---|---|---|---|---|---|---|---|---|---|---|
|  | Mic | Mac | Mic | Mac | Mic | Mac | Mic | Mac | Mic | Mac |
| SyntFunct | 73.9 | 77.8 | 82.1 | 81.3 | 68.0 | 66.5 | 55.9 | 50.3 | 61.4 | 57.3 |
| LogLV | 82.6 | 83.7 | 87.4 | 85.5 | 79.1 | 74.5 | 73.3 | 68.5 | 76.1 | 71.4 |

Table 3: Clustering results using our model to detect alternate linkings (LogLV) against the baseline (SyntFunc).

We can therefore increase cluster purity by clustering only those instances where the model does not indicate an alternation. The results are shown in Table 3. Using less instances while keeping the number of clusters the same will by itself tend to increase performance. To compensate for this, we also report results for the baseline on a reduced dataset. The latter was obtained from the original dataset by randomly removing the same number of instances.[5] By using the model to detect alternations, scores improve over the baseline across the board. We observe performance gains for purity which increases by 8.7% (micro-average; compare Tables 2 and 3). F1 also improves considerably by 13% (micro-average). These results are encouraging indicating that detecting alternate linkings is an important first step towards more accurate role induction.

We also conducted a more detailed error analysis to gain more insight into the behavior of our model. In most cases, alternate linkings where $A1$ occurs in subject position and $A0$ in object position are canonicalized correctly (with 96% and 97% precision, respectively). Half of the detected non-standard linkings involve adjunct roles. Here, the model has much more difficulty with canonicalization and is successful approximately 25% of the time. For example, in the phrase *occur at dawn* the model canonicalizes LOC to ADV, whereas TMP would be the correct function. About 75% of all false negatives are due to core roles and only 25% due to adjunct roles. Many false negatives are due to parser errors, which are reproduced by the model. This indicates overfitting, and indeed many of the false negatives involve infrequent lexical items (e.g., *juxtapose* or *Odyssey*).

Finally, to put our evaluation results into context, we also wanted to compare against Grenager and Manning's (2006) related system. A direct comparison is somewhat problematic due to the use of different datasets and the fact that we induce labels for *all* roles whereas they collapse adjunct roles to a single role. Nevertheless, we made a good-faith effort to evaluate our system using their evaluation setting. Specifically, we ran our system on the same test set, Section 23 of the Penn Treebank (annotated with PropBank roles), using gold standard parses with six clusters for each verb type. Our model achieves a cluster purity score of 90.3% on this dataset compared to 89.7% reported in Grenager and Manning.

## 7 Conclusions

In this paper we have presented a novel framework for unsupervised role induction. We conceptualized the induction problem as one of detecting alternate linkings and finding their canonical syntactic form, and formulated a novel probabilistic model that performs these tasks. The model extends the logistic classifier with latent variables and is trained on parsed output which is used as a noisy target for learning. Experimental results show promise, alternations can be successfully detected and the quality of the induced role clusters can be substantially enhanced.

We argue that the present model could be usefully employed to enhance the performance of other models. For example, it could be used in an active learning context to identify argument instances that are difficult to classify for a supervised or semi-supervised system and would presumably benefit from additional (manual) annotation. Importantly, the framework can incorporate different probabilistic models for detection and canonicalization which we intend to explore in the future. We also aim to embed and test our role induction method within a full SRL system that is also concerned with argument identification. Eventually, we also intend to replace the treebank-trained parser with a chunker.

[5] This was repeated several times to ensure that the results are stable across runs.

## References

Abend, O., R. Reichart, and A. Rappoport. 2009. Unsupervised Argument Identification for Semantic Role Labeling. In *Proceedings of ACL-IJCNLP*. Singapore, pages 28–36.

Berger, A., S. Della Pietra, and V. Della Pietra. 1996. A Maximum Entropy Approach to Natural Language Processing. *Computational Linguistics* 22(1):39–71.

Bishop, C. 2006. *Pattern Recognition and Machine Learning*. Springer.

Bottou, L. 2004. Stochastic Learning. In *Advanced Lectures on Machine Learning*, Springer Verlag, Lecture Notes in Artificial Intelligence, pages 146–168.

Dowty, D. 1991. Thematic Proto Roles and Argument Selection. *Language* 67(3):547–619.

Fillmore, C. J., C. R. Johnson, and M. R. L. Petruck. 2003. Background to FrameNet. *International Journal of Lexicography* 16:235–250.

Fürstenau, H. and M. Lapata. 2009. Graph Aligment for Semi-Supervised Semantic Role Labeling. In *Proceedings of EMNLP*. Singapore, pages 11–20.

Gildea, D. and D. Jurafsky. 2002. Automatic Labeling of Semantic Roles. *Computational Linguistics* 28(3):245–288.

Grenager, T. and C. Manning. 2006. Unsupervised Discovery of a Statistical Verb Lexicon. In *Proceedings of EMNLP*. Sydney, Australia, pages 1–8.

Lapata, M. 1999. Acquiring Lexical Generalizations from Corpora: A Case Study for Diathesis Alternations. In *Proceedings of the 37th ACL*. pages 397–404.

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

Marcus, M., B. Santorini, and M. Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: the Penn Treebank. *Computational Linguistics* 19(2):313–330.

Màrquez, L., X. Carreras, K. Litkowski, and S. Stevenson. 2008. Semantic Role Labeling: an Introduction to the Special Issue. *Computational Linguistics* 34(2):145–159.

McCarthy, D. 2002. Using Semantic Preferences to Identify Verbal Participation in Role Switching Alternations. In *Proceedings of the 1st NAACL*. Seattle, WA, pages 256–263.

McCarthy, D. and A. Korhonen. 1998. Detecting Verbal Participation in Diathesis Alternations. In *Proceedings of COLING/ACL*. Montréal, Canada, pages 1493–1495.

Melli, G., Y. Wang, Y. Liu, M. M. Kashani, Z. Shi, B. Gu, A. Sarkar, and F. Popowich. 2005. Description of SQUASH, the SFU Question Answering Summary Handler for the DUC-2005 Summarization Task. In *Proceedings of the HLT/EMNLP Document Understanding Workshop*. Vancouver, Canada.

Nivre, J., J. Hall, J. Nilsson, G. Eryigit A. Chanev, S. Kübler, S. Marinov, and E. Marsi. 2007. MaltParser: A Language-independent System for Data-driven Dependency Parsing. *Natural Language Engineering* 13(2):95–135.

Padó, S. and M. Lapata. 2009. Cross-lingual Annotation Projection of Semantic Roles. *Journal of Artificial Intelligence Research* 36:307–340.

Palmer, M., D. Gildea, and P. Kingsbury. 2005. The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics* 31(1):71–106.

Pradhan, S. S., W. Ward, and J. H. Martin. 2008. Towards Robust Semantic Role Labeling. *Computational Linguistics* 34(2):289–310.

Schulte im Walde, S. and C. Brew. 2002. Inducing German Semantic Verb Classes from Purely Syntactic Subcategorisation Information. In *Proceedings of the 40th ACL*. Philadelphia, PA, pages 223–230.

Shen, D. and M. Lapata. 2007. Using Semantic Roles to Improve Question Answering. In *Proceedings of the EMNLP-CoNLL*. Prague, Czech Republic, pages 12–21.

Surdeanu, M., S. Harabagiu, J. Williams, and P. Aarseth. 2003. Using Predicate-Argument Structures for Information Extraction. In *Proceedings of the 41st ACL*. Sapporo, Japan, pages 8–15.

Surdeanu, M., R. Johansson, A. Meyers, and L. Màrquez. 2008. The CoNLL-2008 Shared Task on Joint Parsing of Syntactic and Semantic Dependencies. In *Proceedings of the 12th CoNLL*. Manchester, England, pages 159–177.

Sutton, C. and A. McCallum. 2007. An Introduction to Conditional Random Fields for Relational Learning. In L. Getoor and B. Taskar, editors, *Introduction to Statistical Relational Learning*, MIT Press, pages 93–127.

Swier, R. and S. Stevenson. 2004. Unsupervised Semantic Role Labelling. In *Proceedings of EMNLP*. Barcelona, Spain, pages 95–102.

Wu, D. and P. Fung. 2009. Semantic Roles for SMT: A Hybrid Two-Pass Model. In *Proceedings of NAACL HLT 2009: Short Papers*. Boulder, Colorado, pages 13–16.

Yedidia, J., W. Freeman, and Y. Weiss. 2003. Understanding Belief Propagation and its Generalizations. Morgan Kaufmann Publishers Inc., pages 239–269.