# Unsupervised Semantic Role Labelling

**Robert S. Swier** and **Suzanne Stevenson**
Department of Computer Science
University of Toronto
Toronto, Ontario, Canada M5S 3G4
{swier,suzanne}@cs.toronto.edu

Proceedings of EMNLP 2004, 95-102.

## Abstract

We present an unsupervised method for labelling the arguments of verbs with their semantic roles. Our bootstrapping algorithm makes initial unambiguous role assignments, and then iteratively updates the probability model on which future assignments are based. A novel aspect of our approach is the use of verb, slot, and noun class information as the basis for backing off in our probability model. We achieve 50–65% reduction in the error rate over an informed baseline, indicating the potential of our approach for a task that has heretofore relied on large amounts of manually generated training data.

## 1 Introduction

Semantic annotation of text corpora is needed to support tasks such as information extraction and question-answering (e.g., Riloff and Schmelzenbach, 1998; Niu and Hirst, 2004). In particular, labelling the semantic roles of the arguments of a verb (or any predicate), as in (1) and (2), provides crucial information about the relations among event participants.

1. Kiva$_{\langle Experiencer\rangle}$ admires Mats$_{\langle Cause\rangle}$
2. Jo$_{\langle Agent\rangle}$ returned to London$_{\langle Destination\rangle}$

Because of the importance of this task, a number of recent methods have been proposed for automatic semantic role labelling (e.g., Gildea and Jurafsky, 2002; Gildea and Palmer, 2002; Chen and Rambow, 2003; Fleischman et al., 2003; Hacioglu et al., 2003; Thompson et al., 2003). These supervised methods are limited by their reliance on the manually role-tagged corpora of FrameNet (Baker et al., 1998) or PropBank (Palmer et al., 2003) as training data, which are expensive to produce, are limited in size, and may not be representative.

We have developed a novel method of unsupervised semantic role labelling that avoids the need for expensive manual labelling of text, and enables the use of a large, representative corpus. To achieve this, we take a "bootstrapping" approach (e.g., Hindle and Rooth, 1993; Yarowsky, 1995; Jones et al., 1999), which initially makes only the role assignments that are unambiguous according to a verb lexicon. We then iteratively: create a probability model based on the currently annotated semantic roles, use this probability model to assign roles that are deemed to have sufficient evidence, and add the newly labelled arguments to our annotated set. As we iterate, we gradually both grow the size of the annotated set, and relax the evidence thresholds for the probability model, until all arguments have been assigned roles.

To our knowledge, this is the first unsupervised semantic role labelling system applied to general semantic roles in a domain-general corpus. In a similar vein of work, Riloff and colleagues (Riloff and Schmelzenbach, 1998; Jones et al., 1999) used bootstrapping to learn "case frames" for verbs, but their approach has been applied in very narrow topic domains with topic-specific roles. In other work, Gildea (2002) has explored unsupervised methods to discover role-slot mappings for verbs, but not to apply this knowledge to label text with roles.

Our approach also differs from earlier work in its novel use of classes of information in backing off to less specific role probabilities (in contrast to using simple subsets of information, as in Gildea and Jurafsky, 2002). If warranted, we base our decisions on the probability of a role given the verb, the syntactic slot (syntactic argument position), and the noun occurring in that slot. For example, the assignment to the first argument of sentence (1) above may be based on $P(\text{Experiencer}|admire, \text{subject}, Kiva)$. When backing off from this probability, we use statistics over more general classes of information, such as conditioning over the semantic class of the verb instead of the verb itself—for this example, psychological state verbs. Our approach yields a very simple probability model which emphasizes class-based generalizations.

The first step in our algorithm is to use the verb lexicon to determine the argument slots and the roles available for them. In Section 2, we discuss the lexicon we use, and our initial steps of syntactic frame matching and "unambiguous" role assignment. This unambiguous data is leveraged by using those role assignments as the basis for the initial estimates for the probability model described in Section 3. Section 4 presents the algorithm which brings these two components together, iteratively updating the probability estimates as more and more data is labelled. In Section 5, we describe details of the materials and methods used for the experiments presented in Section 6. Our results show a large improvement over an informed baseline. This kind of unsupervised approach to role labelling is quite new, and we conclude with a discussion of limitations and on-going work in Section 7.

## 2 Determining Slots and Role Sets

Previous work has divided the semantic role labelling task into the identification of the arguments to be labelled, and the tagging of each argument with a role (Gildea and Jurafsky, 2002; Fleischman et al., 2003). Our algorithm addresses both these steps. Also, the unsupervised nature of the approach highlights an intermediate step of determining the set of possible roles for each argument. Because we need to constrain the role set as much as possible, and cannot draw on extensive training data, this latter step takes on greater significance in our work.

We first describe the lexicon that specifies the syntactic arguments and possible roles for the verbs, and then discuss our process of argument and role set identification.

### 2.1 The Verb Lexicon

In semantic role labelling, a lexicon is used which lists the possible roles for each syntactic argument of each predicate. Supervised approaches to this task have thus far used the predicate lexicon of FrameNet, or the verb lexicon of PropBank, since each has an associated labelled corpus for training. We instead make use of VerbNet (Kipper et al., 2000), a manually developed hierarchical verb lexicon based on the verb classification of Levin (1993). For each of 191 verb classes, including around 3000 verbs in total, VerbNet specifies the syntactic frames along with the semantic role assigned to each slot of a frame. Throughout the paper we use the term "frame" to refer to a syntactic frame—the set of syntactic arguments of a verb—possibly labelled with roles, as exemplified in the VerbNet entry in Table 1.

While FrameNet uses semantic roles specific to a particular situation (such as Speaker, Message, Addressee), and PropBank uses roles specific to a verb (such as Arg0, Arg1, Arg2), VerbNet uses an intermediate level of thematic roles (such as Agent, Theme, Recipient). These general thematic roles are commonly assumed in linguistic theory, and have some advantages in terms of capturing commonalities of argument relations across a wide range of predicates. It is worth noting that although there are fewer of these thematic roles than the more situation-specific roles of FrameNet, the role labelling task is not necessarily easier: there may be more data per role, but possibly less discriminating data, since each role applies to more general relations. (Indeed, in comparing the use of FrameNet roles to general thematic roles, Gildea and Jurafsky (2002) found very little difference in performance.)

*admire*
**Frames:**
Experiencer V Cause
Experiencer V Cause Prep(in) Oblique
Experiencer V Oblique Prep(for) Cause
**Verbs in same (sub)class:**
[admire, adore, appreciate, cherish, enjoy, ...]

Table 1: A portion of a VerbNet entry.

### 2.2 Frame Matching

We devise a frame matching procedure that uses the verb lexicon to determine, for each instance of a verb, the argument slots and their possible thematic roles. The potential argument slots are subject, object, indirect object, and PP-object, where the latter is specialized by the individual preposition.[1] Given chunked sentences with our verbs, the frame matcher uses VerbNet both to restrict the list of candidate roles for each slot, and to eliminate some of the PP slots that are likely not arguments.

To initialize the candidate roles precisely, we only choose roles from frames in the verb's lexical entry (cf. Table 1) that are the best syntactic matches with the chunker output. We align the slots of each frame with the chunked slots, and compute the portion *%Frame* of frame slots that can be mapped to a chunked slot, and the portion *%Chunks* of chunked slots that can be mapped to the frame. The score for each frame is computed by *%Frame*+*%Chunks*, and only frames having the highest score contribute candidate roles to the chunked slots. An example

[1] As in VerbNet, we assume that when a verb takes a PP argument, the slot receiving the thematic role from the verb is the NP object of the preposition. Also, VerbNet has few verbs that take sentence complements, and for now we do not consider them.

| Possible Frames for V | Extracted Slots SUBJ | OBJ | POBJ | %Frame | %Chunks | Score |
|---|---|---|---|---|---|---|
| Agent V | Agent | | | 100 | 33 | 133 |
| Agent V Theme | Agent | Theme | | 100 | 67 | 167 |
| Instrument V Theme | Instrument | Theme | | 100 | 67 | 167 |
| Agent V Theme P Instrument | Agent | Theme | Instrument | 100 | 100 | 200 |
| Agent V Recipient Theme | Agent | Recipient | | 67 | 67 | 133 |

Table 2: An example of frame matching.

scoring is shown in Table 2.

This frame matching step is very restrictive and greatly reduces potential role ambiguity. Many syntactic slots receive only a single candidate role, providing the initial unambiguous data for our bootstrapping algorithm. Some slots receive *no* candidate roles, which is an error for argument slots but which is correct for adjuncts. The reduction of candidate roles in general is very helpful in lightening the load on the probability model, but note that it may also cause the correct role to be omitted. In future work, we plan to explore other possible methods of selecting roles from the frames, such as choosing candidates from all frames, or setting a threshold value on the matching score.

## 3 The Probability Model

Once slots are initialized as above, our algorithm uses an iteratively updated probability model for role labelling. The probability model predicts the role for a slot given certain conditioning information. We use a backoff approach with three levels of specificity of probabilities. If a candidate role fails to meet the threshold of evidence (counts towards that probability) for a given level, we backoff to the next level. For any given slot, we use the most specific level that reaches the evidence threshold for any of the candidates. We only use information at a single level to compare candidates for a single slot.

We assume the probability of a role for a slot is independent of other slots; we do not ensure a consistent role assignment across an instance of a verb.

### 3.1 The Backoff Levels

Our most specific probability uses the exact combination of verb, slot, and noun filling that slot, yielding $P(r|v,s,n)$.[2]

[2]We use only the head noun of potential arguments, not the full NP, in our probability model. Our combination of slot plus head word provides similar information (head of argument and its syntactic relation to the verb) to that captured by the features of Gildea and Jurafsky (2002) or Thompson et al. (2003).

For our first backoff level, we introduce a novel way to generalize over the verb, slot, and noun information of $P(r|v,s,n)$. Here we use a linear interpolation of three probabilities, each of which: (1) drops one source of conditioning information from the most specific probability, and (2) generalizes a second source of conditioning information to a class-based conditioning event. Specifically, we use the following probability formula:

$$\lambda_1 P_1(r|v,sc) + \lambda_2 P_2(r|v,nc) + \lambda_3 P_3(r|vc,s)$$

where $sc$ is slot class, $nc$ is noun class, $vc$ is verb class, and the individual probabilities are (currently) equally weighted (i.e., all $\lambda_i$'s have a value of $1/3$).

Note that all three component probabilities make use of the verb or its class information. In $P_1$, the noun component is dropped, and the slot is generalized to the appropriate slot class. In $P_2$, the slot component is dropped, and the noun is generalized to the appropriate noun class. Although it may seem counterintuitive to drop the slot, this helps us capture generalizations over "alternations," in which the same semantic argument may appear in different syntactic slots (as in <u>The ice</u> *melted* and *The sun melted* <u>*the ice*</u>). In $P_3$, again the noun component is dropped, but in this case the verb is generalized to the appropriate verb class. Each type of class is described in the following subsection.

The last backoff level simply uses the probability of the role given the slot class, $P(r|sc)$. The backoff model is summarized in Figure 1. We use maximum likelihood estimates (MLE) for each of the probability formulas.

### 3.2 Classes of Information

For slots, true generalization to a class only occurs for the prepositional slots, all of which are mapped to a single PP slot class. All other slots—subject, object, and indirect object—each form their own singleton slot class. Thus, $P_1$ differs from $P(r|v,s,n)$ by dropping the noun, and by treating all prepositional slots as the same slot. This formula allows us to generalize over a slot regardless of the

$$\boxed{P(r|v,s,n)} \longrightarrow \boxed{\begin{array}{c}\lambda_1 P_1(r|v,sc)\\ +\\ \lambda_2 P_2(r|v,nc)\\ +\\ \lambda_3 P_3(r|vc,s)\end{array}} \longrightarrow \boxed{P(r|sc)}$$

Figure 1: The backoff model.

particular noun, and preposition if there is one, used in the instance.

Classes of nouns in the model are given by the WordNet hierarchy. Determining the appropriate level of generalization for a noun is an open problem (e.g., Clark and Weir, 2002). Currently, we use a cut through WordNet including all the top categories, except for the category "entity"; the latter, because of its generality, is replaced in the cut by its immediate children (Schulte im Walde, 2003). Given a noun argument, all of its ancestors that appear in this cut are used as the class(es) for the noun. (Credit for a noun is apportioned equally across multiple classes.) Unknown words placed in a separate category. This yields a noun classification system that is very coarse and that does not distinguish between senses, but which is simple and computationally feasible. $P_2$ thus captures consistent relations between a verb and a class of nouns, regardless of the slot in which the noun occurs.

Verb classes have been shown to be very important in capturing generalizations across verb behaviour in computational systems (e.g., Palmer, 2000; Merlo and Stevenson, 2001). In semantic role labelling using VerbNet, they are particularly relevant since the classes are based on a commonality of role-labelled syntactic frames (Kipper et al., 2000). The class of a verb in our model is its VerbNet class that is compatible with the current frame. When multiple classes are compatible, we apportion the counts uniformly among them. For probability $P_3$, then, we generalize over all verbs in a class of the target verb, giving us much more extensive data over relevant role assignments to a particular slot.

## 4 The Bootstrapping Algorithm

We have described the frame matcher that produces a set of slots with candidate role lists (some unambiguous), and our backoff probability model. All that remains is to specify the parameters that guide the iterative use of the probability model to assign roles.

The evidence count for each of the conditional probabilities refers to the number of times we have observed the conjunction of its conditioning events. For example, for $P(r|v,s,n)$, this is the number of times the particular combination of verb, slot, and noun have been observed. For a probability to be used, its evidence count must reach a given threshold, $minEvidence$.

The "goodness" of a role assignment is determined by taking the log of the ratio between the probabilities of the top two candidates for a slot (when the evidence of both meet $minEvidence$) (e.g., Hindle and Rooth, 1993). A role is only assigned if the log likelihood ratio is defined and meets a threshold; in this case, the candidate role with highest probability is assigned to the slot. (Note that in the current implementation, we do not allow re-labelling: an assigned label is fixed.) In the algorithm, the log ratio threshold is initially set high and gradually reduced until it reaches 0. In the case of remaining ties, we assign the role for which $P(r|sc)$ is highest.

Because our evidence count and log ratio restrictions may not be met even when we have a very good candidate for a slot, we reduce the evidence count threshold to the minimum value of 1 when the log ratio threshold reaches 1.[3] By this point, we assume competitor candidates have been given sufficient opportunity to amass the relevant counts.

Algorithm 1 shows the bootstrapping algorithm.

**Algorithm 1** Bootstrapping Algorithm

**Frame Matching, Slot Initialization:**

1: Perform Frame Matching to determine the slots to be labelled, along with their candidate lists of roles.

2: Let $A$ be the set of annotated slots; $A = \emptyset$.
Let $U$ be the set of unannotated slots, initially all slots.
Let $N$ be the set of newly annotated slots; $N = \emptyset$.

3: Add to $N$ each slot whose role assignment is unambiguous—whose candidate list has one element.
Set $U$ to $U - N$ and set $A$ to $A + N$ (where $-$ and $+$ remove/add elements of the second set from/to the first).

**Probability Model Application:**

```
repeat
  repeat
    (Re)compute the probability model, using counts over
    the items in A.
    Add to N all slots in U for which:
      –at least two candidates meet the evidence count
      threshold for a given probability level (see Figure 1);
      and
      –the log ratio between the two highest probability can-
      didates meets the log ratio threshold.
    For each slot in N, assign the highest probability role.
    Set U to U − N and set A to A + N.
  until N = ∅
  Decrement the log ratio threshold.
  Adjust evidence count threshold if log ratio threshold is 1.
until log ratio threshold = 0
Resolve ties and terminate.
```

[3] We also allow cases in which the log ratio is undefined to be assigned at this point—this occurs when only one of multiple candidates has evidence.

## 5 Materials and Methods

### 5.1 Verbs, Verb Classes and Roles

For the initial set of experiments, we chose 54 target verbs from three top-level VerbNet classes: preparing-26.3, transfer_mesg-37.1, and contribute-13.2. We looked for classes that contained a large number of medium to high frequency verbs displaying a variety of interesting properties, such as having ambiguous (or unambiguous) semantic roles given certain syntactic constructions, or having ambiguous semantic role assignments that could (or alternatively, could not) be distinguished by knowledge of verb class.

From the set of target verbs, we derived an extended verb set that comprises all of the original target verbs as well as any verb that shares a class with one of those target verbs. This gives us a set of 1159 verbs to observe in total, and increases the likelihood that some verb class information is available for each of the possible classes of the target verbs. Observing the entire extended set also provides more data for our probability estimators that do not use verb class information.

We have made several changes to the semantic roles as given by VerbNet. First, selectional restrictions such as [+Animate] are removed since our coarse model of noun class does not allow us to reliably determine whether such restrictions are met. Second, a few semantic distinctions that are made in VerbNet appeared to be too fine-grained to capture, so we map these to a more coarse-grained subset of the VerbNet roles. For instance, the role Actor is merged with Agent, and Patient with Theme. We are left with a set of 16 roles: Agent, Amount, Attribute, Beneficiary, Cause, Destination, Experiencer, Instrument, Location, Material, Predicate, Recipient, Source, Stimulus, Theme, Time. Of these, 13 actually occur in our target verb classes.

### 5.2 The Corpus and Preprocessing

Our corpus consists of a random selection of 20% of the sentences in the British National Corpus (BNC Reference Guide, 2000). This corpus is processed by the chunker of Abney (1991), from whose output we can identify the probable head words of verb arguments with some degree of error. For instance, distant subjects are often not found, and PPs identified as arguments are often adjuncts. To reduce the number of adjuncts, we ignore dates and any PPs that are not known to (possibly) introduce an argument to one of the verbs in our extended set.

### 5.3 Validation and Test Data

We extracted two sets of sentences: a validation set consisting of 5 random examples of each target verb, and a test set, consisting of 10 random examples of each target verb. The data sets were chunked as above, and the role for each potential argument slot was labelled by two human annotators, choosing from the simplified role set allowed by each verb according to VerbNet. A slot could also be labelled as an adjunct, or as "bad" (incorrectly chunked). Agreement between the two annotators was high, yielding a kappa statistic of 0.83. After performing the labelling task individually, the annotators reconciled their responses (in consultation with a third annotator) to yield a set of human judgements used for evaluation.

### 5.4 Setting the Bootstrapping Parameters

In our development experiments, we tried an evidence count threshold of either the mean or median over all counts of a particular conjunction of conditioning events. (For example, for $P(r|v, s, n)$, this is the mean or median count across all combinations of verb, slot, and noun.) The more lenient median setting worked slightly better on the validation set, and was retained for our test experiments. We also experimented with initial starting values of 2, 3, and 8 for the log likelihood ratio threshold. An initial setting of 8 showed an improvement in performance, as lower values enabled too many early role assignments, so we used the value of 8 in our test experiments. In all experiments, a decrement of .5 was used to gradually reduce the log likelihood ratio threshold.

## 6 Experimental Results

Of over 960K slots we extracted from the corpus, 120K occurred with one of 54 target verbs. Of these, our validation data consisted of 278 slots, and our test data of 554 slots. We focus on the analysis of test data; the pattern on the validation data was nearly identical in all respects.

The target slots fall into several categories, depending on the human judgements: argument slots, adjunct slots, and "bad" slots (chunking errors). We report detailed analysis over the slots identified as arguments. We also report overall accuracy if adjunct and "bad" slots are included in the slots to be labelled. This comparison is similar to that made by Gildea and Jurafsky (2002) and others, either using arguments as delimited in the FrameNet corpus, or having to automatically locate argument boundaries.[4] Furthermore, we report results over individ-

[4] The comparison is not identical: in the case of manually

ual slot classes (subject, object, indirect object, and PP object), as well as over all slots.

### 6.1 Evaluation Measures and Comparisons

We report results after the "unambiguous" data is assigned, and at the end of the algorithm, when no more slots can be labelled. At either of these steps it is possible for some slots to have been assigned and some to remain unassigned. Rather than performing a simple precision/recall analysis, we report a finer grained elaboration that gives a more precise picture of the results. For the assigned slots, we report percent correct (of *total*, not of assigned) and percent incorrect. For the unassigned slots, we report percent "possible" (i.e., slots whose candidate list contains the correct role) and percent "impossible" (i.e., slots whose candidate list does not contain the correct role—and which may in fact be empty). All these percent figures are out of all argument slots (for the first set of results), and out of all slots (for the second set); see Table 3. Correctness is determined by the human judgements on the chunked slots, as reported above.

Using our notion of slot class, we compare our results to a baseline that assigns all slots the role with the highest probability for that slot class, $P(r|sc)$. When using general thematic roles, this is a more informed baseline than $P(r|v)$, as used in other work.

We are using a very different verb lexicon, corpus, and human standard than in previous research. The closest work is that of Gildea and Jurafsky (2002) which maps FrameNet roles to a set of 18 thematic roles very similar to our roles, and also operates on a subset of the BNC (albeit manually rather than randomly selected). We mention the performance of their method where appropriate below. However, our results are compared to human annotation of chunked data, while theirs (and other supervised results) are compared to manually annotated full sentences. Our percentage correct values therefore do not take into account argument constituents that are simply missed by the chunker.

### 6.2 Results on Argument Slots

Table 3 summarizes our results. In this section, we focus on argument slots as identified by our human judges (the first panel of results in the table). There are a number of things to note. First, our performance on these slots is very high, 90.1% correct at the end of the algorithm, with 7.0% incorrect, and only 2.9% left unassigned. (The latter have null candidate lists.) This is a 56% reduction in error rate over the baseline. Second, we see that even after the initial unambiguous role assignment step, the algorithm achieves close to the baseline percent correct. Furthermore, over 96% of the initially assigned roles are correct. This means that much of the work in narrowing down the candidate lists is actually being preformed during frame matching. It is noteworthy that such a simple method of choosing the initial candidates can be so useful, and it would seem that even supervised methods might benefit from employing such an explicit use of the lexicon to narrow down role candidates for a slot.

After unambiguous role assignment, about 21% of the test data remains unassigned (116 slots). Of these 116 slots, 100 have a non-null candidate list. These 100 are assigned by our iterative probability model, so we are especially interested in the results on them. We find that 76 of these 100 are assigned correctly (accounting for the 13.7% increase to 90.1%), and 24 are assigned incorrectly, yielding a 76% accuracy for the probability model portion of our algorithm on identified argument slots.

Moreover, we also find that all specificity levels of the probability model (see Figure 1) are employed in making these decisions—about a third of the decisions are made by each level. This indicates that while there is sufficient data in many cases to warrant using the exact probability formula $P(r|v, s, n)$, the class-based generalizations we propose prove to be very useful to the algorithm.

As a point of comparison, the supervised method of Gildea and Jurafsky (2002) achieved 82.1% accuracy on identified arguments using general thematic roles. However, they had a larger and more varied target set, consisting of 1462 predicates from 67 FrameNet frames (classes), which makes their task harder than ours. We are aware that our test set is small compared to supervised approaches, which have a large amount of labelled data available. However, our almost identical results across the validation and test sets indicates consistent behaviour that may generalize to a larger test set, at least on similar classes of verbs.

### 6.3 Differences Among Slot Classes

When using general thematic roles with a small set of verb classes, the probability used for the baseline, $P(r|sc)$, works very well for subjects and objects (which are primarily Agents and Themes, respectively, for our verbs). Indeed, when we examine each of the slot classes individually, we find that, for subjects and objects, the percent correct

delimited arguments, others train, as well as test, only on such arguments. In our approach, all previously annotated slots are used in the iterative training of the probability model. Thus, even when we report results on argument slots only, adjunct and "bad" slots may have induced errors in their labelling.

| Role Assignments | | Identified Arguments Baseline | Algorithm "Unambig" | Final | All Target Slots Baseline | Algorithm "Unambig" | Final |
|---|---|---|---|---|---|---|---|
| Assigned | Correct | 77.3 | 76.4 | 90.1 | 63.7 | 75.9 | 87.2 |
| | Incorrect | 22.7 | 2.7 | 7.0 | 36.3 | 6.8 | 10.4 |
| Unassigned | Possible | 0 | 17.1 | 0 | 0 | 14.1 | 0 |
| | Impossible | 0 | 3.8 | 2.9 | 0 | 3.1 | 2.4 |

Table 3: Evaluation of test data on 554 identified arguments (see Section 6.2) and on all 672 target slots (see Section 6.4).

achieved by the algorithm is indistinguishable from the baseline (both are around 93%, for both subjects and objects). For PP objects, on the other hand, the baseline is only around 11% correct, while we achieve 78.5% correct, a 76% reduction in error rate. Clearly, when more roles are available, even $P(r|sc)$ becomes a weak predictor.[5]

We could just assign the default role for subjects and objects when using general thematic roles, but we think this is too simplistic. First, when we broaden our range of verb classes, subjects and objects will have more possible roles. As we have seen with PPs, when more roles are available, the performance of a default role degrades. Second, although we achieve the same correctness as the baseline, our algorithm does not simply assign the dominant role in these cases. Some subjects are assigned Theme, while some objects are assigned Recipient or Source. These roles would never be possible in these slots if a default assignment were followed.

### 6.4 Results Including All Target Slots

We also consider our performance given frame matching and chunking errors, which can lead to adjuncts or even "bad" constituents being labelled. Only arguments should be labelled, while non-arguments should remain unlabelled. Of 98 slots judged to be adjuncts, 19 erroneously are given labels. Including the adjunct slots, our percent correct goes from 90.1% to 88.7%. Of the 20 "bad" slots, 12 were labelled. Including these, correctness is reduced slightly further, to 87.2%, as shown in the second panel of results in Table 3. The error rate reduction here of 65% is higher than on arguments only, because the baseline always labels (in error) adjuncts and "bad" slots. (Gildea and Jurafsky (2002) achieved 63.6% accuracy when having to identify arguments for thematic roles, though note again that this is on a much larger and more general test set. Also, although we take into account errors on identified chunks that are not arguments, we are are not counting chunker errors of missing arguments.)

As others have shown (Gildea and Palmer, 2002), semantic role labelling is more accurate with better preprocessing of the data. However, we also think our algorithm may be extendable to deal with many of the adjunct cases we observed. Often, adjuncts express time or location; while not argument roles, these do express generalizable semantic relations. In future work, we plan to explore the notion of expanding our frame matching step to go beyond VerbNet by initializing potential adjuncts with appropriate roles.

## 7 Conclusions and Future Work

Using an unsupervised algorithm for semantic role labelling, we have achieved 90% correct on identified arguments, well over an informed baseline of 77%, and have achieved 87% correct on all slots (64% baseline). On PP objects, our conservative role assignment shows promise at leaving adjuncts unlabelled. However, PP objects also have the lowest performance (of 78% correct on identified arguments, compared to 93% for subjects or objects). More work is required on our frame matching approach to determine appropriate roles for PP objects given the specification in the lexicon, which (in the case of VerbNet) often over-constrains the allowable prepositions for a slot.

Although these results are promising, they are only a first step in demonstrating the potential of the approach. We need to test more verbs, from a wider variety of verb classes (or even a different kind of predicate classification, such as FrameNet), to determine the generalizability of our findings. Using FrameNet would also have the advantage of providing large amounts of labelled test data for our evaluation. We also hope to integrate some processing of adjunct roles, rather than limiting ourselves to the specified arguments.

[5] Due to the rarity of indirect object slots in the chunker output, the test data included no such slots. The validation set included one, which the algorithm correctly labelled.

A unique aspect of our method is the probability model, which is novel in its generalizations over verb, slot, and noun classes for role labelling. However, these have room for improvement—our noun classes are coarse, and prepositions clearly have the potential to be divided into more informative subclasses, such as spatial or time relations. Our ongoing work is investigating better class models to make the backoff process even more effective.

## Acknowledgments

We gratefully acknowledge the support of NSERC of Canada. We also thank Martha Palmer for providing us with the VerbNet data, Eric Joanis for help with the chunker, Vivian Tsang and Ryan North for helpful discussion, and two anonymous reviewers.

## References

S. Abney. 1991. Parsing by chunks. In R. Berwick, S. Abney, and C. Tenny, editors, *Principle-Based Parsing*. Kluwer Academic Publishers.

C. F. Baker, C. J. Fillmore, and J. B. Lowe. 1998. The Berkeley FrameNet Project. In *Proceedings of COLING-ACL*.

BNC Reference Guide. 2000. *Reference Guide for the British National Corpus (World Edition)*. http://www.hcu.ox.ac.uk/BNC, second edition.

J. Chen and O. Rambow. 2003. Use of deep linguistic features for the recognition and labeling of semantic arguments. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing*.

S. Clark and D. Weir. 2002. Probability estimation using a semantic hierarchy. *Computational Linguistics*, 28(2):187–206.

M. Fleischman, N. Kwon, and E. Hovy. 2003. Maximum entropy models for FrameNet classification. In *Proc. of the Conf. on Empirical Methods in Natural Language Processing*.

D. Gildea. 2002. Probabilistic models of verb-argument structure. In *Proc. of the 19th International Conference on Computational Linguistics (COLING-02)*, p. 308–314.

D. Gildea and D. Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 23(3):245–288.

D. Gildea and M. Palmer. 2002. The necessity of syntactic parsing for predicate argument recognition. In *Proc. of the 40th Annual Conf. of the Assoc. for Computational Linguistics*, p. 239–246.

K. Hacioglu, S. Pradhan, W. Ward, J. H. Martin, and D. Jurafsky. 2003. Semantic role labeling by tagging syntactic chunks. In *Proc. of the 8th Conf. on Computational Natural Language Learning*.

D. Hindle and M. Rooth. 1993. Structural ambiguity and lexical relations. *Computational Linguistics*, 19(1):103–120.

R. Jones, A. McCallum, K. Nigam, and E. Riloff. 1999. Bootstrapping for text learning tasks. In *IJCAI-99 Workshop on Text Mining: Foundations, Techniques and Applications*.

K. Kipper, H. T. Dang, and M. Palmer. 2000. Class based construction of a verb lexicon. In *Proc. of the 17th National Conference on Artificial Intelligence (AAAI-2000)*.

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

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

Y. Niu and G. Hirst. 2004. Analysis of semantic classes in medical text for question answering. In *Workshop on Question Answering in Restricted Domains, 42nd Annual Meeting of the Assoc. for Computational Linguistics*.

M. Palmer. 2000. Consistent criteria for sense distinctions. *Special Issue of Computers and the Humanities, SENSEVAL98: Evaluating Word Sense Disambiguation Systems*, 34(1–2).

M. Palmer, D. Gildea, and P. Kingsbury. 2003. The Proposition Bank: An annotated corpus of semantic roles. Submitted to *Computational Linguistics*.

E. Riloff and M. Schmelzenbach. 1998. An empirical approach to conceptual case frame acquisition. In *Proc. of the 6th Workshop on Very Large Corpora*.

S. Schulte im Walde. 2003. Experiments on the choice of features for learning verb classes. In *Proc. of the 10th Conf. of the European Chapter of the Assoc. for Computational Linguistics*.

C. Thompson, R. Levy, and C. Manning. 2003. A generative model for FrameNet semantic role labeling. In *Proc. of the Fourteenth European Conf. on Machine Learning (ECML-03)*.

D. Yarowsky. 1995. Unsupervised word sense disambiguation methods rivaling supervised methods". In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, p. 189–196.