# Exploiting a Verb Lexicon in Automatic 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 HLT/EMNLP 2005

## Abstract

We develop an unsupervised semantic role labelling system that relies on the direct application of information in a predicate lexicon combined with a simple probability model. We demonstrate the usefulness of predicate lexicons for role labelling, as well as the feasibility of modifying an existing role-labelled corpus for evaluating a different set of semantic roles. We achieve a substantial improvement over an informed baseline.

## 1 Introduction

Intelligent language technologies capable of full semantic interpretation of domain-general text remain an elusive goal. However, statistical advances have made it possible to address core pieces of the problem. Recent years have seen a wealth of research on one important component of semantic interpretation—automatic role labelling (e.g., Gildea and Jurafsky, 2002; Pradhan et al., 2004; Hacioglu et al., 2004, and additional papers from Carreras and Marquez, 2004). Such work aims to annotate each constituent in a clause with a semantic tag indicating the role that the constituent plays with respect to the target predicate, as in (1):

(1) [Yuka]$_{Agent}$ [whispered]$_{Pred}$ to [Dar]$_{Recipient}$

Semantic role labelling systems address a crucial first step in the automatic extraction of semantic relations from domain-general text, taking us closer to the goal of comprehensive semantic mark-up.

Most work thus far on domain-general role labelling depends on supervised learning over statistical features extracted from a hand-labelled corpus. The reliance on such a resource—one in which the arguments of each predicate are manually identified and assigned a semantic role—limits the portability of such methods to other languages or even to other genres of corpora.

In this study, we explore the possibility of using a verb lexicon, rather than a hand-labelled corpus, as the primary resource in the semantic role labelling task. Perhaps because of the focus on what can be gleaned from labelled data, existing supervised approaches have made little use of the additional knowledge available in the predicate lexicon associated with the labelled corpus. By contrast, we exploit the explicit knowledge of the role assignment possibilities for each verb within an existing lexicon. Moreover, we utilize a very simple probability model within a highly efficient algorithm.

We use VerbNet (Kipper et al., 2000), a computational lexicon which lists the possible semantic role assignments for each of its verbs. Our algorithm extracts automatically parsed arguments from a corpus, and assigns to each a list of the compatible roles according to VerbNet. Arguments which are given only a single role possibility are considered to have been assigned an unambiguous role label. This set of arguments constitutes our *primary-labelled* data, which serves as the noisy training data for a simple probability model which is then used to label the remaining (role ambiguous) arguments.

This method has several advantages, the foremost of which is that it eliminates the dependence on a role labelled corpus, a very expensive resource to produce. Of course, a verb lexicon is also an expensive resource, but one that is highly reusable across a range of NLP tasks. Moreover, the approach points at some potentially useful information that current

supervised methods have failed to exploit. Even if one has access to an annotated corpus for training, our work shows that directly calling on additional information from the lexicon itself may prove useful in restricting the possible labels for an argument.

The method has disadvantages as well. The information available in a predicate lexicon is less directly applicable to building a learning model. Inevitably, our results are noisier than in a supervised approach which has access to a labelled sample of what it must produce. Still, the method shows promise: on unseen test data, the system yields an F-measure of .83 on labelling of correctly extracted arguments, compared to an informed baseline of .74, and an F-measure of .65 (compared to .52) on the overall identification and labelling task. The latter is well below the best supervised performance of about .80 on similar tasks, but it must be emphasized that it is achieved with a simple probability model and without the use of hand-labelled data. We view this as a starting point by which to demonstrate the utility of deriving more explicit knowledge from a predicate lexicon, which can be later extended through the use of additional probabilistic features.

We face a methodological challenge arising from the particular choice of VerbNet for the prototyping of our method: the lexicon has no associated semantic role labelled corpus. While this underscores the need for approaches which do not rely on such a resource, it also means that we lack a labelled sample of data against which to evaluate our results. To address this, we use the existing labelled corpus of FrameNet (Baker et al., 1998), and develop a mapping for converting the FrameNet roles to corresponding VerbNet roles. Our mapping method demonstrates the possibility of leveraging existing resources to support the development of role labelling systems based on verb lexicons that do not have an associated hand-labelled corpus.

## 2 VerbNet Roles and the Role Mapping

Before describing our labelling algorithm, we first briefly introduce the semantic role information available in VerbNet, and describe how we map FrameNet roles to VerbNet roles.

*whisper*
**Frames:**
Agent V
Agent V Prep(+dest) Recipient
Agent V Topic
**Verbs in same (sub)class:**
[bark, croon, drone, grunt, holler, ...]

Figure 1: A portion of a VerbNet entry.

### 2.1 The VerbNet Lexicon

VerbNet is a manually developed hierarchical lexicon based on the verb classification of Levin (1993). For each of almost 200 classes containing a total of 3000 verbs, VerbNet specifies the syntactic frames along with the semantic role assigned to each argument position of a frame.[1] Figure 1 shows an example VerbNet entry. The thematic roles used in VerbNet are more general than the situation-specific roles of FrameNet. For example, the roles Speaker, Message, and Addressee of a Communication verb such as *whisper* in FrameNet would be termed Agent, Topic, and Recipient in VerbNet. These coarser-grained roles are often assumed in linguistic theory, and have some advantages in terms of capturing commonalities of argument relations across a wide range of predicates.

### 2.2 Mapping FrameNet to VerbNet Roles

As noted, VerbNet lacks a corpus of example role assignments against which to evaluate a role labelling based upon it. We create such a resource by adapting the existing FrameNet corpus. We formulate a mapping between FrameNet's larger role set and VerbNet's much smaller one, and create a new corpus with our mapped roles substituted for the original roles in the FrameNet corpus.

We perform the mapping in three steps. First we use an existing mapping between the semantically-specific roles in FrameNet and a much smaller intermediate set of 39 semantic roles which subsume all FrameNet roles.[2] The associations in this mapping are straightforward—e.g., the Place role for Abusing verbs and the Area role for Operate-vehicle verbs are both mapped to Location.

[1] Throughout the paper we use the term "frame" to refer to a syntactic frame—a configuration of syntactic arguments of a verb—possibly labelled with roles, as in Figure 1.

[2] This mapping was provided by Roxana Girju, UIUC.

Second, from this intermediate set we create a simple mapping to the set of 22 VerbNet roles. Some roles are unaffected by the mapping (e.g., Cause alone in the intermediate set maps to Cause in the VerbNet set). Other roles are merged (e.g., Degree and Measure both map to Amount). Moreover, some roles in FrameNet (and the intermediate set) must be mapped to more than one VerbNet role. For example, an Experiencer role in FrameNet is considered Experiencer by some VerbNet classes, but Agent by others. In such cases, our mappings in this step must be specific to the VerbNet class.

In this second step, some roles have no subsuming VerbNet role, because FrameNet provides roles for a wider variety of relations. For example, both FrameNet and the intermediate role set contain a Manner role, which VerbNet does not have. We create a catch-all label, "NoRole," to which we map eight such intermediate roles: Condition, Manner, Means, Medium, Part-Whole, Property, Purpose, and Result. These phrases labelled NoRole are adjuncts—constituents not labelled by VerbNet.

In the third step of our mapping, some of the roles in VerbNet—such as Theme and Topic, Asset and Amount—which appear to be too-fine grained for us to distinguish reliably, are mapped to a more coarse-grained set of VerbNet roles. The final set consists of 16 roles: Agent, Amount, Attribute, Beneficiary, Cause, Destination, Experiencer, Instrument, Location, Material, Predicate, Recipient, Source, Stimulus, Theme and Time; plus the NoRole label.

## 3 The Frame Matching Process

A main goal of our system is to demonstrate the usefulness of predicate lexicons for the role labelling task. The primary way that we apply the knowledge in our lexicon is via a process we call *frame matching*, adapted from Swier and Stevenson (2004). The automatic frame matcher aligns arguments extracted from an automatically parsed sentence with the frames in VerbNet for the target verb in the sentence. The output of this process is a highly constrained set of candidate roles (possibly of size one) for each potential argument. The resulting singleton sets constitute a (noisy) role assignment for their corresponding arguments, forming our primary-labelled data. This data is then used to train a probability model, described in Section 4, which we employ to label the remaining arguments (those having more than one candidate role).

### 3.1 Initialization of Candidate Roles

The frame matcher construes extracted arguments from the parsed sentence as being in one of the four main types of syntactic positions (or *slots*) used by VerbNet frames: subject, object, indirect object, and PP-object.[3] Additionally, we specialize the latter by the individual preposition, such as "object of *for*." For the first three slot types, alignment between the extracted arguments and the frames is relatively straightforward. An extracted subject would be aligned with the subject position in a VerbNet frame, for instance, and the subject role from the frame would be listed as a possible label for the extracted subject.

The alignment of PP-objects is similar to that of the other slot types, except that we add an additional constraint that the associated prepositions must match. For PP-object slots, VerbNet frames often provide an explicit list of allowable prepositions. Alternatively, the frame may specify a required semantic feature such as `+path` or `+loc`. In order for an extracted PP-object to align with one of these frame slots, its associated preposition must be included in the list provided by the frame, or have the specified feature. To determine the latter, we manually create lists of prepositions that we judge to have each of the possible semantic features.

In general, this matching procedure assumes that frames describing a syntactic argument structure similar to that of the parsed sentence are more likely to correctly describe the semantic roles of the extracted arguments. Thus, the frame matcher only chooses roles from frames that are the best syntactic matches with the extracted argument set. This is achieved by adopting the scoring method of Swier and Stevenson (2004), in which we compute the portion *%Frame* of frame slots that can be mapped to an extracted argument, and the portion *%Sent* of extracted arguments from the sentence that can be mapped to the frame. The score for each frame is given by *%Frame*+*%Sent*, and only frames having the highest score contribute candidate roles to the

[3] Since VerbNet has very few verbs with sentential complements, we do not consider them for now.

| Possible Frames for Verb V | Extracted Slots | | %Frame | %Sent | Score |
|---|---|---|---|---|---|
| | SUBJ | OBJ | | | |
| Agent V | Agent | | 100 | 50 | 150 |
| **Agent V Theme** | **Agent** | **Theme** | **100** | **100** | **200** |
| **Instrument V Theme** | **Instrument** | **Theme** | **100** | **100** | **200** |
| Agent V Recipient Theme | Agent | Theme | 67 | 100 | 167 |

Table 1: An example of frame matching.

extracted arguments. An example scoring is shown in Table 1. Note that two of the frames are tied for the highest score of 200, resulting in two possible roles for the subject (Agent and Instrument), and Theme as the only possible role for the object.

As mentioned, this frame matching step is very restrictive, and it greatly reduces role ambiguity. Many potential arguments receive only a single candidate role, providing the primary-labelled data we use to train our probability model. 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 subsequent load on the probability model to be applied next, but note that it may also cause the correct role to be omitted. We experiment with choosing roles from the frames that are the best syntactic matches, and from all possible frames.

### 3.2 Adjustments to the Role Mapping

We further extend the frame matcher, which has extensive knowledge of VerbNet, for the separate task of helping to eliminate some of the inconsistencies that are introduced by our role mapping procedure. This is a process that applies concurrently with the initialization of candidate roles described above, but only affects the gold standard labelling of evaluation data.[4]

For instance, FrameNet assigns the role Side2 to the object of the preposition *with* occurring with the verb *brawl*. Side2 is mapped to Theme by our role mapping; however, in VerbNet, *brawl* does not accept Theme as the object of *with*. Our mapping thus creates a target (i.e., gold standard) label in the evaluation data that is inconsistent with VerbNet. Since there is no possibility of the role labeller assigning a label that matches such a target, this unfairly raises the task difficulty. However, since *brawl* does accept Theme in another slot, it is not an option to entirely eliminate this role in the mapping for the verb. Instead, we use our frame matcher to verify that each target role generated by our mapping from FrameNet is allowed by VerbNet in the relevant slot. If the target role is not allowed, then it is converted to NoRole in the evaluation set. Constituents labelled as NoRole are not considered target arguments, and it is correct for the system to not assign labels in these cases.

The NoRole conversions help to ensure that our gold standard evaluation data is consistent with our lexicon, but the method does have limitations. For instance, some of the arguments which the system fails to extract might have had their target role changed to NoRole if they were properly extracted. Additionally, in some cases a target role is converted to NoRole when there is an actual role that VerbNet would have assigned instead.

## 4 The Probability Model

Once argument slots are initialized with sets of possible roles, the algorithm uses a probability model to label slots having two or more possibilities. Since our primary goal is to demonstrate how much can be accomplished through the frame matcher, we compare a number of very simple probability models:

- $\mathbf{P}(\mathbf{r}|\mathbf{v},\mathbf{s})$: the probability of a role given the target verb and the slot; the latter includes subject, object, indirect object, and prepositional object, where each PP slot is specialized by the identity of the preposition;
- $\mathbf{P}(\mathbf{r}|\mathbf{s})$: the probability of a role given the slot;
- $\mathbf{P}(\mathbf{r}|\mathbf{sc})$: the probability of a role given the slot class, in which all prepositional slots are treated together.

[4] Of course, the fact that the frame matcher "sees" the evaluation set as part of its dual duties is not allowed to influence its assignment of candidate roles.

Each probability model predicts a role given certain conditioning information, with maximum likelihood estimates determined by the primary-labelled data directly resulting from the frame matching step.[5]

We also compare one non-probabilistic model to resolve the same set of ambiguous cases:

- **Default assignment**: candidate roles for ambiguous slots are ignored; the four slot classes of subject, object, indirect object and PP-object are assigned the roles Agent, Theme, Recipient, and Location, respectively.

These are the most likely roles assigned by the frame matcher over our development data.

For comparison, we also apply the iterative algorithm developed by Swier and Stevenson (2004), using the same bootstrapping parameters. The method uses backoff over three levels of specificity of probabilities.

## 5 Materials and Methods

### 5.1 The Target Verbs

For ease of comparison, we use the same verbs as in Swier and Stevenson (2004), except that we measure performance over a much larger superset of verbs. In that work, a core set of 54 target verbs are selected to represent a variety of classes with interesting role ambiguities, and the system is evaluated against only those verbs. An additional 1105 verbs—all verbs sharing at least one class with the target verbs—are also labelled, in order to provide more data for the probability estimations. Here, we consider our system's performance over the 1159 target verbs that consist of the union of these two sets of verbs.

### 5.2 The Corpus and Preprocessing

The majority of sentences in FrameNet II are taken from the British National Corpus (BNC Reference Guide, 2000). Our development and test data consists of a percentage of these sentences. For some experiments, these sentences are then merged with a random selection of additional sentences from the BNC in order to provide more training data for the probability estimations. We evaluate performance only on FrameNet sentences that include our target verbs.

All of our corpus data was parsed using the Collins parser (Collins, 1999). Next, we use TGrep2 (Rohde, 2004) to automatically extract from the parse trees the constituents forming potential arguments of the target verbs. For each verb, we label as the subject the lowest NP node, if it exists, that is immediately to the left of a VP node which dominates the verb. Other arguments are identified by finding sister NP or PP nodes to the right of the verb. Heads of noun phrases are identified using the method of Collins (1999), which primarily chooses the rightmost noun in the phrase that is not inside a prepositional phrase or subordinate clause. Error may be introduced at each step of this preprocessing—the sentence may be misparsed, some arguments (such as distant subjects) may not be extracted, or the wrong word may be found as the phrase head.

### 5.3 Validation and Test Data

A random selection of 30% of the preprocessed FrameNet data is set aside for testing, and another random 30% is used for development and validation. For experiments involving additional BNC data, each 30% of the FrameNet sentences is embedded in a random selection of 20% of the BNC. We selected these percentages to yield a sufficient amount of data for experimentation, while reserving some unseen data for future work. The FrameNet portion of the validation set includes 515 types of our target verbs (across 161 VerbNet classes) in 4300 sentences, and contains a total of 6636 target constituents—i.e., constituents that receive a valid VerbNet role as their gold standard label, not NoRole. The test set includes 517 of the target verbs (from 163 classes) in 4308 sentences, yielding 6705 target constituents.[6]

To create an evaluation set, we map the manually annotated FrameNet roles in the corpus to VerbNet roles (or NoRole), as described in Sections 2.2 and 3.2. We use this role information to calculate performance: the system should assign roles matching the target VerbNet roles, and make no assignment when the target is NoRole.

[5] Note that we assume the probability of a role for a slot is independent of other slots—that is, we do not ensure a consistent role assignment to all arguments across an instance of a verb.

[6] The verbs appearing in the validation and test sets occur respectively across 161 and 165 FrameNet classes (what in FrameNet are called "frames").

### 5.4 Methods of Argument Identification

One of the decisions we face is how to evaluate the identification of extracted arguments generated by the system against the manually annotated target arguments provided by FrameNet. We try two methods, the most strict of which is to require full-phrase agreement: an extracted argument and a target argument must cover exactly the same words in the sentence in order for the argument to be considered correctly extracted. This means, for instance, that a prepositional phrase incorrectly attached to an extracted object would render the object incompatible with the target argument, and any system label on it would be counted as incorrect. This evaluation method is commonly used in other work (e.g., Carreras and Marquez, 2004).

The other method we use is to require that only the head of an extracted argument and a target argument match. This latter method helps to provide a fuller picture of the range of arguments found by the system, since there are fewer near-misses caused by attachment errors. Since heads of phrases are often the most semantically relevant part of an argument, labels on heads provide much of the same information as labels on whole phrases. For these reasons, we use head matching for most of our experiments below. For comparison, however, we provide results based on full-phrase matching as well.

## 6 Experimental Results

### 6.1 Experimental Setup

We evaluate our system's performance on several aspects of the overall role labelling task; all results are given in terms of F-measure, $2PR/(P+R)$.[7] The first task is argument identification, in which constituents considered by our system to be arguments (i.e., those that are extracted and labelled) are evaluated against actual target arguments. The second task is labelling extracted arguments, which evaluates the labelling of only those arguments that were correctly extracted. Last is the overall role labelling task, which evaluates the system on the combined tasks of identification and labelling of all target arguments.

We compare our results to an informed baseline that has access to the same set of extracted arguments as does the frame matcher. The baseline labels all extracted arguments using the default role assignments described in Section 4.

In addition to experiments in which we employ various methods of resolving ambiguous assignments, we also evaluate the system with varying types and amounts of training data, and with two alternate methods for choosing frames from which to draw candidate roles.

### 6.2 Evaluation of Probability Models

We first evaluate our system with the three very simple probability models, as well as the non-probabilistic default assignment, to determine roles for the extracted arguments that the frame matcher considers to be ambiguous. We also report results after only the frame matcher has been applied, to indicate how much work is being done by it alone. Because we have constructed the frame matcher to be highly restrictive in assigning candidate roles to extracted arguments, a large number (about 62%) become primary-labelled data and so do not require resolution of ambiguous roles. Only about 16% of our extracted arguments have role ambiguities, and about 22% (many of which are adjuncts) do not receive any candidates and remain unlabelled.

| Task: | Id. | Lab. | Id. + Lab. |
|---|---|---|---|
| Baseline | .80 | .74 | .52 |
| FM + $P(r\|sc)$ | .83 | .83 | .65 |
| FM + $P(r\|s)$ | .83 | .84 | .65 |
| FM + $P(r\|v,s)$ | .83 | .78 | .61 |
| FM + Dflt. Assgnmt. | .83 | .82 | .64 |
| FM only | .83 | .76 | .60 |

As shown in the table, all models perform equally well on identification, which is determined by the frame matcher (FM); i.e., any extracted argument receiving one or more candidate roles is "identified" as an argument. Performance is somewhat above the baseline, which must label *all* extracted arguments. For the task of labelling correctly extracted arguments and for the combined task, the simplest probability models, $P(r|sc)$ and $P(r|s)$, perform about the same. On the combined task, they achieve .13 above the informed baseline, indicating the effectiveness of such simple models when combined with the frame matcher. The more specific model, $P(r|v,s)$, performs less well, and may be over-fitting on this relatively small amount of train-

[7] In each case, $P$ and $R$ are close in value.

ing data.

Two observations indicate the power of the frame matcher. First, even using the non-probabilistic default assignments to resolve ambiguous roles substantially outperforms the baseline (and indeed performs quite close to the best results, since the default role assignment is often the same as that chosen by the probability models). Importantly, the baseline uses the *same* default assignments, but without the benefit of the frame matcher to further narrow down the possible arguments. Second, the frame matcher alone achieves .60 F-measure on the combined task, not far below the performance of the best models. These results show that once arguments have been extracted, much of the labelling work is performed by the frame matcher's careful application of lexical information.

Henceforth we consider the use of the frame matcher plus $P(r|sc)$ as our basic system, since this is our simplest model, and no other outperforms it.

### 6.3 Evaluation of Training Methods

In our above experiments, the probabilistic models are trained only on primary-labelled data from the frame matcher run on the FrameNet data. We would like to determine whether using either more data or less noisy data may improve results. To provide more data, we ran the frame matcher on the additional 20% of the BNC. This provides almost 600K more sentences containing our target verbs, yielding a much higher amount of primary-labelled data. To provide less-noisy data, we trained the probability models on manually annotated target labels from system-identified arguments in 1000 sentences. While fewer sentences are used, all arguments in the training data are guaranteed to have a correct role assignment, in contrast to the primary-labelled data output by the frame matcher. (We chose 1000 sentences as an upper bound on an amount of data that could be relatively easily annotated by human judges.)

| Training Data: | Prim.-lab. FN | Prim.-lab. BNC | 1K sents annot'd |
|---|---|---|---|
| Baseline | .52 | | |
| FM + $P(r\|sc)$ | .65 | .65 | .65 |
| FM + $P(r\|v,s)$ | .61 | .62 | .63 |

For our basic model, $P(r|sc)$, these variations in training data do not affect performance. Only the most specific model, $P(r|v,s)$, shows improvement when trained on more data or on manually annotated data, although it still does not perform as well as the simplest model. Because the models only choose from among candidate roles selected by the frame matcher, differences in the learned probability estimations must be quite large to have an effect. At least for the simplest model, these estimations do not vary with a larger corpus or one lacking in noise. However, the increase in performance seen here for the more specific model, albeit small, may indicate that richer probability models may require more or cleaner training data.

### 6.4 Evaluation of Frame Choice

| | "Best" frames | All Frames |
|---|---|---|
| Baseline | .52 | |
| FM + $P(r\|sc)$ | .65 | .63 |

The frame matcher has been shown to shoulder much of the responsibility in our system, and it is worth considering variations in its operation. For example, by having the frame matcher only choose roles from the frames that are the best syntactic matches to the sentence, role ambiguity is minimized at the cost of possibly excluding the correct role. To determine whether we may do better by relying more on the probability model and less on the frame matcher, we instead include role candidates from all frames in a verb's lexical entry. The effect of this choice is more role ambiguity, decreasing the number of primary-labelled slots by roughly 30%. We see that performance using $P(r|sc)$ is slightly worse with the greater ambiguity admitted by using all frames, indicating the benefit of precise selection of candidate roles.

### 6.5 Differing Argument Evaluation Methods

| | Heads | Full Phrase |
|---|---|---|
| Baseline | .52 | .49 |
| FM + $P(r\|sc)$ | .65 | .61 |

As mentioned, for most of our evaluations we match the arguments extracted by the system to the target arguments via a match on phrase heads, since head labels provide much useful semantic information. When we instead require that the extracted arguments match the targets exactly, the number of correctly extracted arguments falls from about 80%

of the roughly 6700 targets to about 74%, due to increased parsing difficulty. As expected, this results in both the system and the baseline having performance decreases on the overall task.

## 7 Related Work

Most role labelling systems have required hand-labelled training data. Two exceptions are the subcategorization frame based work of Atserias et al. (2001) and the bootstrapping labeller of Swier and Stevenson (2004), but both are evaluated on only a small number of verbs and arguments. In related unsupervised tasks, Riloff and colleagues have learned "case frames" for verbs (e.g., Riloff and Schmelzenbach, 1998), while Gildea (2002) has learned role-slot mappings (but does not apply the knowledge for the labelling task).

Other role labelling systems have also relied on the extraction of much more complex features or probability models than we adopt here. As a point of comparison, we apply the iterative backoff model from Swier and Stevenson (2004), trained on 20% of the BNC, with our frame matcher and test data. The backoff model achieves an F-measure of .63, slightly below the performance of .65 for our simplest probability model, which uses less training data and takes far less time to run (minutes rather than hours).

In general, it is not possible to make direct comparisons between our work and most other role labellers because of differences in corpora and role sets, and, perhaps more significantly, differences in the selection of target arguments. However, the best supervised systems, using automatic parses to identify full argument phrases in PropBank, achieve about .82 on the task of identifying and labelling arguments (Pradhan et al., 2004). Though this is higher than our performance of .61 on full phrase arguments, our system does not require manually annotated data.

## 8 Conclusion

In this work, we employ an expensive but highly reusable resource—a verb lexicon—to perform role labelling with a simple probability model and a small amount of unsupervised training data. We outperform similar work that uses much more data and a more complex model, showing the benefit of exploiting lexical information directly. To achieve performance comparable to that of supervised methods may require human filtering or augmentation of the initial labelling. However, given the expense of producing a large semantically annotated corpus, even such "human in the loop" approaches may lead to a decrease in overall resource demands. We use such a corpus for evaluation purposes only, modifying it with a role mapping to correspond to our lexicon. We thus demonstrate that such existing resources can be bootstrapped for lexicons lacking an associated annotated corpus.

## Acknowledgments

We gratefully acknowledge the support of NSERC of Canada. We also thank Afsaneh Fazly, who assisted with much of our corpus pre-processing.

## References

J. Atserias, L. Padró, and G. Rigau. 2001. Integrating multiple knowledge sources for robust semantic parsing. In *Proc. of the International Conf. on Recent Advances in NLP*.

C. F. Baker, C. J. Fillmore, and J. B. Lowe. 1998. The Berkeley FrameNet Project. In *Proc. of COLING-ACL*, p. 86–90.

BNC Reference Guide. 2000. *Reference Guide for the British National Corpus (World Edition)*, second edition.

X. Carreras and L. Marquez, editors. 2004. *CoNLL-04 Shared Task*.

M. Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

D. Gildea. 2002. Probabilistic models of verb-argument structure. In *Proc. of the 19th International CoNLL*, p. 308–314.

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

K. Hacioglu, S. Pradhan, W. Ward, J. H. Martin, and D. Jurafsky. 2004. Semantic role labeling by tagging syntactic chunks. In *Proc. of the 8th CoNLL*, p. 110–113.

K. Kipper, H. T. Dang, and M. Palmer. 2000. Class based construction of a verb lexicon. In *Proc. of the 17th AAAI Conf.*

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

S. Pradhan, W. Ward, K. Hacioglu, J. Martin, and D. Jurafsky. 2004. Shallow semantic parsing using support vector machines. In *Proc. of HLT/NAACL*.

E. Riloff and M. Schmelzenbach. 1998. An empirical approach to conceptual case frame acquisition. In *Proc. of the 6th WVLC*.

D. L. T. Rohde. 2004. TGrep2 user manual ver. 1.11. http://tedlab.mit.edu/~dr/Tgrep2.

R. Swier and S. Stevenson. 2004. Unsupervised semantic role labelling. In *Proc. of the 2004 Conf. on EMNLP*, p. 95–102.