Frame-Semantic Role Labeling with Heterogeneous Annotations

Meghana Kshirsagar*  Sam Thomson*  Nathan Schneider†
Jaime Carbonell*  Noah A. Smith*  Chris Dyer*

*School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA
†School of Informatics, University of Edinburgh, Edinburgh, Scotland, UK

Abstract

We consider the task of identifying and labeling the semantic arguments of a predicate that evokes a FrameNet frame. This task is challenging because there are only a few thousand fully annotated sentences for supervised training. Our approach augments an existing model with features derived from FrameNet and PropBank and with partially annotated exemplars from FrameNet. We observe a 4% absolute increase in $F_1$ versus the original model.

1 Introduction

Paucity of data resources is a challenge for semantic analyses like frame-semantic parsing (Gildea and Jurafsky, 2002; Das et al., 2014) using the FrameNet lexicon (Baker et al., 1998; Fillmore and Baker, 2009). Given a sentence, a frame-semantic parse maps word tokens to frames they evoke, and for each frame, finds and labels its argument phrases with frame-specific roles. An example appears in figure 1.

In this paper, we address this argument identification subtask, a form of semantic role labeling (SRL), a task introduced by Gildea and Jurafsky (2002) using an earlier version of FrameNet. Our contribution addresses the paucity of annotated data for training using standard domain adaptation techniques. We exploit three annotation sources:

- the frame-to-frame relations in FrameNet, by using hierarchical features to share statistical strength among related roles ($\S$3.2),
- FrameNet’s corpus of partially-annotated ex-emplar sentences, by using “frustratingly easy” domain adaptation ($\S$3.3), and

- a PropBank-style SRL system, by using guide features ($\S$3.4).²

These expansions of the training corpus and the feature set for supervised argument identification are integrated into SEMAFOR (Das et al., 2014), the leading open-source frame-semantic parser for English. We observe a 4% $F_1$ improvement in argument identification on the FrameNet test set, leading to a 1% $F_1$ improvement on the full frame-semantic parsing task. Our code and models are available at http://www.ark.cs.cmu.edu/SEMAFOR/.

2 FrameNet

FrameNet represents events, scenarios, and relationships with an inventory of frames (such as

<table>
<thead>
<tr>
<th>Frame</th>
<th>exemplar sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>AM-ADV</td>
<td>do you want me to hold off until I finish July and August?</td>
</tr>
<tr>
<td>A1</td>
<td>the people really want us to stay the course and finish the job.</td>
</tr>
<tr>
<td>A3</td>
<td>wait.v</td>
</tr>
</tbody>
</table>


²Preliminary experiments training on PropBank annotations mapped to FrameNet via SemLink 1.2.2c (Bonial et al., 2013) hurt performance, likely due to errors and coverage gaps in the mappings.
we consider two kinds of frame-to-frame relations: with several kinds of frame-to-frame relations. Inheritance: There are 4,138 The FrameNet lexicon is organized as a network, and 589 Subframe links among role types in FrameNet 1.5. Prior work has considered various ways of grouping role labels together in order to share statistical strength. Matsubayashi et al. (2009) observed small gains from using the Inheritance relationships and also from grouping by the role name (SEMAFOR already incorporates such features). Johansson (2012) reports improvements in SRL for Swedish, by exploiting relationships between both frames and roles. Baldewein et al. (2004) learn latent clusters of roles and role-fillers, reporting mixed results. Our approach is described in §3.2.

2.1 Hierarchy
The FrameNet lexicon is organized as a network, with several kinds of frame-to-frame relations linking pairs of frames and (subsets of) their arguments (Ruppenhofer et al., 2010). In this work, we consider two kinds of frame-to-frame relations: Inheritance: E.g., ROBBERY inherits from COMMITTING_CRIME, which inherits from MISDEED. Crucially, roles in inheriting frames are mapped to corresponding roles in inherited frames: ROBBERY_Perpetrator links to COMMITTING_CRIME_Perpetrator, which links to MISDEED_Wrongdoer, and so forth. Subframe: This indicates a subevent within a complex event. E.g., the CRIMINAL_PROCESS frame groups together subframes ARREST, ARRAIGNMENT and TRIAL. CRIMINAL_PROCESS_Defendant, for instance, is mapped to ARREST_SUSPECT, TRIAL_Defendant, and SENTENCING_Convict.

We say that a parent of a role is one that has either the Inheritance or Subframe relation to it. There are 4,138 Inheritance and 589 Subframe links among role types in FrameNet 1.5.

Prior work has considered various ways of grouping role labels together in order to share statistical strength. Matsubayashi et al. (2009) observed small gains from using the Inheritance relationships and also from grouping by the role name (SEMAFOR already incorporates such features). Johansson (2012) reports improvements in SRL for Swedish, by exploiting relationships between both frames and roles. Baldewein et al. (2004) learn latent clusters of roles and role-fillers, reporting mixed results. Our approach is described in §3.2.

2.2 Annotations
Statistics for the annotations appear in table 1.

Full-text (FT): This portion of the FrameNet corpus consists of documents and has about 5,000 sentences for which annotators assigned frames and arguments to as many words as possible. Beginning with the SemEval-2007 shared task on FrameNet analysis, frame-semantic parsers have been trained and evaluated on the full-text data (Baker et al., 2007; Das et al., 2014). The full-text documents represent a mix of genres, prominently including travel guides and bureaucratic reports about weapons stockpiles.

Exemplars: To document a given predicate, lexicographers manually select corpus examples and annotate them only with respect to the predicate in question. These singly-annotated sentences from FrameNet are called lexicographic exemplars. There are over 140,000 sentences containing argument annotations and relative to the FT dataset, these contain an order of magnitude more frame annotations and over two orders of magnitude more sentences. As these were manually selected, the rate of overt arguments per frame is noticeably higher than in the FT data. The exemplars formed the basis of early studies of frame-semantic role labeling (e.g., Gildea and Jurafsky, 2002; Thompson et al., 2003; Fleischman et al., 2003; Litkowski, 2004; Kwon et al., 2004). Exemplars have not yet been exploited successfully to improve role labeling performance on the more realistic FT task.4

2.3 PropBank
PropBank (PB; Palmer et al., 2005) is a lexicon and corpus of predicate–argument structures that takes a shallower approach than FrameNet. FrameNet frames cluster lexical predicates that evoke sim-
ilar kinds of scenarios In comparison. PropBank frames are purely lexical and there are no formal relations between different predicates or their roles. PropBank’s sense distinctions are generally coarser-grained than FrameNet’s. Moreover, FrameNet lexical entries cover many parts of speech, while PropBank focuses on verbs and (as of recently) eventive noun and adjective predicates. An example with PB annotations is shown in figure 2.

3 Model

We use the model from SEMAFOR (Das et al., 2014), detailed in §3.1, as a starting point. We experiment with techniques that augment the model’s training data (§3.3) and feature set (§3.2, §3.4).

3.1 Baseline

In SEMAFOR, the argument identification task is treated as a structured prediction problem. Let the classification input be a dependency-parsed sentence x, the token(s) p constituting the predicate in question, and the frame f evoked by p (as determined by frame identification). We use the heuristic procedure described by (Das et al., 2014) for extracting candidate argument spans for the predicate; call this spans(x, p, f). spans always includes a special span denoting an empty or non-overt role, denoted φ. For each candidate argument a ∈ spans(x, p, f) and each role r, a binary feature vector φ(a, x, p, f, r) is extracted. We use the feature extractors from (Das et al., 2014) as a baseline, adding additional ones in our experiments (§3.2–§3.4). Each a is given a real-valued score by a linear model:

\[ \text{score}_a(a | x, p, f, r) = w^T \phi(a, x, p, f, r) \] (1)

The model parameters w are learned from data (§4).

Prediction requires choosing a joint assignment of all arguments of a frame, respecting the constraints that a role may be assigned to at most one span, and spans of overt arguments must not overlap. Beam search, with a beam size of 100, is used to find this arg max.5

3.2 Hierarchy Features

We experiment with features shared between related roles of related frames in order to capture statistical generalizations about the kinds of arguments seen in those roles. Our hypothesis is that this will be beneficial given the small number of training examples for individual roles.

All roles that have a common parent based on the Inheritance and Subframe relations will share a set of features in common. Specifically, for each base feature φ which is conjoined with the role r in the baseline model (φ ∧ "role=r"), and for each parent r’ of r, we add a new copy of the feature that is the base feature conjoined with the parent role, (φ ∧ "parent_role=r’"). We experimented with using more than one level of the hierarchy (e.g., grandparents), but the additional levels did not improve performance.

3.3 Domain Adaptation and Exemplars

Daumé (2007) proposed a feature augmentation approach that is now widely used in supervised domain adaptation scenarios. We use a variant of this approach. Let DEx denote the exemplars training data, and Dft denote the full text training data. For every feature φ(a, x, p, f, r) in the base model, we add a new feature φEx(·) that fires only if φ(·) fires and x ∈ DEx. The intuition is that each base feature contributes both a “general” weight and a “domain-specific” weight to the model; thus, it can exhibit a general preference for specific roles, but this general preference can be fine-tuned for the domain. Regularization encourages the model to use the general version over the domain-specific, if possible.

3.4 Guide Features

Another approach to domain adaptation is to train a supervised model on a source domain, make predictions using that model on the target domain, then use those predictions as additional features while training a new model on the target domain. The source domain model is effectively a form of pre-processing, and the features from its output are known as guide features (Johansson, 2013; Kong et al., 2014).6

In our case, the full text data is our target domain, and PropBank and the exemplars data are our source domains, respectively. For PropBank, we run the SRL system of Illinois Curator 1.1.4 (Pun-
yakanok et al., 2008) on verbs in the full-text data. For the exemplars, we train baseline SEMAFOR on the exemplars and run it on the full-text data.

We use two types of guide features: one encodes the role label predicted by the source model, and the other indicates that a span $a$ was assigned some role. For the exemplars, we use an additional feature to indicate that the predicted role matches the role being filled.

4 Learning

Following SEMAFOR, we train using a local objective, treating each role and span pair as an independent training instance. We have made two modifications to training which had negligible impact on full-text accuracy, but decreased training time significantly:\footnote{With SEMAFOR’s original features and training data, the result of the above changes is that full-text $F_1$ decreases from 59.3\% to 59.1\%, while training time (running optimization to convergence) decreases from 729 minutes to 82 minutes.}

- We use the online optimization method AdaDelta (Zeiler, 2012) with minibatches, instead of the batch method L-BFGS (Liu and Nocedal, 1989). We use minibatches of size 4,000 on the full text data, and 40,000 on the exemplar data.
- We minimize squared structured hinge loss instead of a log-linear loss. Let $(x, p, f, r, a)$ be the $i$th training example. Then the squared hinge loss is given by $L_w(i) = \left(\max_{a'} \left\{ w^T \phi(a', x, p, f, r) + 1_{\{a' \neq a\}} \right\} - w^T \phi(a, x, p, f, r)\right)^2$.

We learn $w$ by minimizing the $\ell_2$-regularized average loss on the dataset:

$$w^* = \arg \min_w \frac{1}{N} \sum_{i=1}^N L_w(i) + \frac{1}{2} \lambda \|w\|_2^2 \quad (2)$$

5 Experimental Setup

We use the same FrameNet 1.5 data and train/test splits as Das et al. (2014). Automatic syntactic dependency parses from MSTParserStacked (Martins et al., 2008) are used, as in Das et al. (2014).

Preprocessing. Out of 145,838 exemplar sentences, we removed 4,191 sentences which had no role annotations. We removed sentences that appeared in the full-text data. We also merged spans which were adjacent and had the same role label.

<table>
<thead>
<tr>
<th>Training Configuration (Features)</th>
<th>Model Size</th>
<th>P (%)</th>
<th>R (%)</th>
<th>F1 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT (Baseline)</td>
<td>1.1</td>
<td>65.6</td>
<td>53.8</td>
<td>59.1</td>
</tr>
<tr>
<td>FT (Hierarchy)</td>
<td>1.9</td>
<td>67.2</td>
<td>54.8</td>
<td>60.4</td>
</tr>
<tr>
<td>Exemplars guide → FT</td>
<td>1.2</td>
<td>65.2</td>
<td>55.9</td>
<td>60.2</td>
</tr>
<tr>
<td>FT+Exemplars (Basic)</td>
<td>5.0</td>
<td>66.0</td>
<td>58.2</td>
<td>61.9</td>
</tr>
<tr>
<td>FT+Exemplars (DA)</td>
<td>5.8</td>
<td>65.7</td>
<td>59.0</td>
<td>62.2</td>
</tr>
<tr>
<td>PB-SRL guide → FT</td>
<td>1.2</td>
<td>65.0</td>
<td>54.8</td>
<td>59.5</td>
</tr>
</tbody>
</table>

Combining the best methods

| PB-SRL guide → FT+Exemplars     | 5.5        | 67.4  | 58.8  | 62.8   |
| FT+Exemplars (Hierarchy)        | 9.3        | 66.0  | 60.4  | 63.1   |

Table 2: Argument identification results on the full-text test set. Model size is in millions of features.

Hyperparameter tuning. We determined the stopping criterion and the $\ell_2$ regularization parameter $\lambda$ by tuning on the FT development set, searching over the following values for $\lambda$: $10^{-5}, 10^{-7}, 10^{-9}, 10^{-12}$.

Evaluation. A complete frame-semantic parsing system involves frame identification and argument identification. We perform two evaluations: one assuming gold-standard frames are given, to evaluate argument identification alone; and one using the output of the system described by Hermann et al. (2014), the current state-of-the-art in frame identification, to demonstrate that our improvements are retained when incorporated into a full system.

6 Results

Argument Identification. We present precision, recall, and $F_1$-measure microaveraged across the test instances in table 2, for all approaches. The evaluation used in Das et al. (2014) assesses both frames and arguments; since our focus is on SRL, we only report performance for arguments, rendering our scores more interpretable. Under our argument-only evaluation, the system of Das et al. (2014) gets 59.3\% $F_1$.

The first block shows baseline performance. The next block shows the benefit of FrameNet hierarchy features (+1.2\% $F_1$). The third block shows that using exemplars as training data, especially with domain adaptation, is preferable to using them as guide features (2.8\% $F_1$ vs. 0.9\% $F_1$). PropBank SRL as guide features offers a small (0.4\% $F_1$) gain.

The last two rows of table 2 show the performance upon combining the best approaches. Both use full-text and exemplars for training; the first uses PropBank SRL as guide features, and the second adds hierarchy features. The best result is the
latter, gaining 3.95% \( F_1 \) over the baseline.

**Role-level evaluation.** Figure 3(b) shows \( F_1 \) per frame element, for the baseline and the three best models. Each x-axis value is one role, sorted by decreasing frequency (the distribution of role frequencies is shown in figure 3(a)). For frequent roles, performance is similar; our models achieve gains on rarer roles.

**Full system.** When using the frame output of Hermann et al. (2014), \( F_1 \) improves by 1.1%, from 66.8% for the baseline, to 67.9% for our combined model (from the last row in table 2).

### 7 Conclusion

We have empirically shown that auxiliary semantic resources can benefit the challenging task of frame-semantic role labeling. The significant gains come from the FrameNet exemplars and the FrameNet hierarchy, with some signs that the PropBank scheme can be leveraged as well.

We are optimistic that future improvements to lexical semantic resources, such as crowdsourced lexical expansion of FrameNet (Pavlick et al., 2015) as well as ongoing/planned changes for PropBank (Bonial et al., 2014) and SemLink (Bonial et al., 2013), will lead to further gains in this task. Moreover, the techniques discussed here could be further explored using semi-automatic mappings between lexical resources (such as UBY; Gurevych et al., 2012), and correspondingly, this task could be used to extrinsically validate those mappings.

Ours is not the only study to show benefit from heterogeneous annotations for semantic analysis tasks. Feizabadi and Pádó (2015), for example, successfully applied similar techniques for SRL of implicit arguments. \(^9\) Ultimately, given the diversity of semantic resources, we expect that learning from heterogeneous annotations in different corpora will be necessary to build automatic semantic analyzers that are both accurate and robust.

### Acknowledgments

The authors are grateful to Dipanjan Das for his assistance, and to anonymous reviewers for their helpful feedback. This research has been supported by the Richard King Mellon Foundation and DARPA grant FA8750-12-2-0342 funded under the DEFT program.

### References


Claire Bonial, Kevin Stowe, and Martha Palmer. 2013. Renewing and revising SemLink. In *Proc. of the...*

\(^9\) They applied frustratingly easy domain adaptation to learn from FrameNet along with a PropBank-like dataset of nominal frames.
2nd Workshop on Linked Data in Linguistics (LDL-2013): Representing and linking lexicons, terminologies and other language data, pages 9–17. Pisa, Italy.


