SEMAFOR 1.0: A Probabilistic Frame-Semantic Parser

Dipanjan Das[*†] Nathan Schneider[*†]
Desai Chen[†] Noah A. Smith[*†]

[*]Language Technologies Institute
[†]School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue,
Pittsburgh, PA 15213, USA

{dipanjan@cs, nschneid@cs, desaic@andrew, nasmith@cs}.cmu.edu

CMU-LTI-10-001

April 1, 2010

Abstract

An elaboration on (Das et al., 2010), this report formalizes frame-semantic parsing as a structure prediction problem and describes an implemented parser that transforms an English sentence into a frame-semantic representation. SEMAFOR 1.0 finds words that evoke FrameNet frames, selects frames for them, and locates the arguments for each frame. The system uses two feature-based, discriminative probabilistic (log-linear) models, one with latent variables to permit disambiguation of new predicate words. The parser is demonstrated to significantly outperform previously published results and is released for public use.

1 Introduction

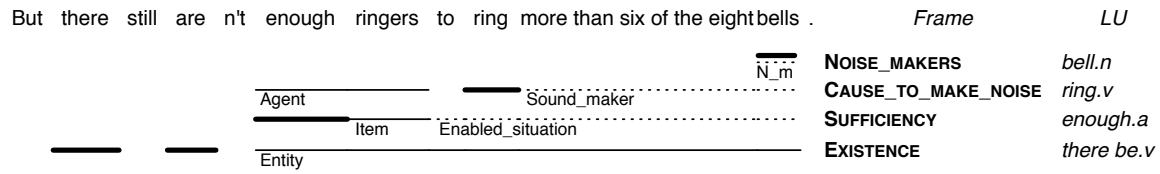

Figure 1: A sentence from PropBank and the SemEval'07 training data, and a partial depiction of gold FrameNet annotations. Each frame is a row below the sentence (ordered for readability). Thick lines indicate targets that evoke frames; thin solid/dotted lines with labels indicate arguments. "N_m" under *bells* is short for the Noise_maker role of the NOISE_MAKERS frame—it is a **denoted frame element** because it is also the target. The last row indicates that *there...are* is a discontinuous target. In PropBank, the verb *ring* is the only annotated predicate for this sentence, and it is not related to other predicates with similar meanings.

FrameNet (Fillmore et al., 2003) is a rich linguistic resource containing considerable information about lexical and predicate-argument semantics in English. Grounded in the theory of frame semantics (Fillmore, 1982), it suggests—but does not formally define—a semantic representation that blends word-sense disambiguation and semantic role labeling.

In this report, we present a computational and statistical model for frame-semantic parsing, the problem of extracting from text semantic predicate-argument structures such as those shown in Fig. 1. We aim to predict a frame-semantic representation as a *structure*, not as a pipeline of classifiers. We use a probabilistic framework that cleanly integrates the FrameNet lexicon and (currently very limited) available training data. Although our models often involve strong independence assumptions, the probabilistic framework we adopt is highly amenable to future extension through new features, relaxed independence assumptions, and semisupervised learning. Some novel aspects of our current approach include a latent-variable model that permits disambiguation of words not in the FrameNet lexicon, a unified model for finding and labeling arguments, and a precision-boosting constraint that forbids arguments of the same predicate to overlap. Our parser, named SEMAFOR,[1] achieves the best published results to date on the SemEval'07 FrameNet task (Baker et al., 2007).

2 Resources and Task

We consider frame-semantic parsing resources.

2.1 FrameNet Lexicon

The FrameNet lexicon is a taxonomy of manually identified general-purpose **frames** for English.[2] Listed in the lexicon with each frame are several lemmas (with part of speech) that can denote the frame or some aspect of it—these are called **lexical units** (LUs). In a sentence, word or phrase tokens that evoke a frame are known as **targets**. The set of LUs listed for a frame in FrameNet may not be exhaustive; we may see a target in new

[1] Semantic Analyzer of Frame Representations

[2] Like the SemEval'07 participants, we used FrameNet v. 1.3 (http://framenet.icsi.berkeley.edu).

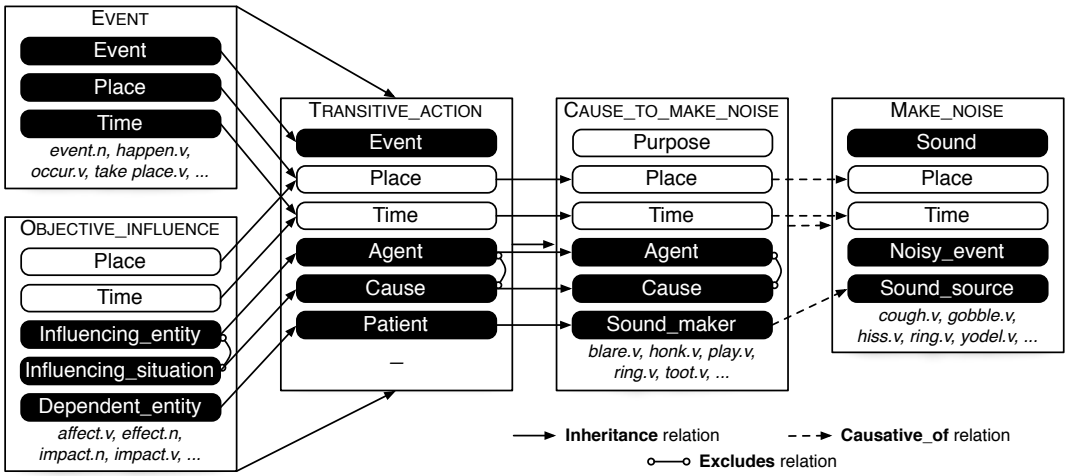

Figure 2: Partial illustration of frames, roles, and LUs related to the CAUSE_TO_MAKE_NOISE frame, from the FrameNet lexicon. "Core" roles are filled ovals. Non-core roles (such as Place and Time) as unfilled ovals. No particular significance is ascribed to the ordering of a frame's roles in its lexicon entry (the selection and ordering of roles above is for illustative convenience). CAUSE_TO_MAKE_NOISE defines a total of 14 roles, many of them not shown here.

data that does not correspond to an LU for the frame it evokes. Each frame definition also includes a set of frame elements, or **roles**, corresponding to different aspects of the concept represented by the frame, such as participants, props, and attributes. We use the term **argument** to refer to a sequence of word tokens annotated as filling a frame role. Fig. 1 shows an example sentence from the training data with annotated targets, LUs, frames, and role-argument pairs. The FrameNet lexicon also provides information about relations between frames and between roles (e.g., INHERITANCE). Fig. 2 shows a subset of the relations between three frames and their roles.

Accompanying most frame definitions in the FrameNet lexicon is a set of lexicographic **exemplar sentences** (primarily from the British National Corpus) annotated for that frame. Typically chosen to illustrate variation in argument realization patterns for the frame in question, these sentences only contain annotations for a single frame. We found that using exemplar sentences directly to train our models hurt performance as evaluated on SemEval'07 data, even though the number of exemplar sentences is an order of magnitude larger than the number of sentences in our training set (§2.2). This is presumably because the exemplars are neither representative as a sample nor similar to the test data. Instead, we make use of these exemplars in features (§4.2).

2.2 Data

Our training, development, and test sets consist of documents annotated with frame-semantic structures for the SemEval'07 task, which we refer to collectively as the **SemEval'07 data**.[3] For the most part, the frames and roles used in annotating these documents were defined in the FrameNet lexicon, but there are some exceptions for which the annotators defined supplementary frames and roles; these are included in the

[3]The full-text annotations and other resources for the 2007 task are available at http://framenet.icsi.berkeley.edu/semeval/FSSE.html.

FRAMENET LEXICON V. 1.3		
lexical entries	**exemplars**	
	counts	*coverage*
8379 LUs	139K sentences, 3.1M words	70% LUs
795 frames	1 frame annotation / sentence	63% frames
7124 roles	285K overt arguments	56% roles

Table 1: Snapshot of lexicon entries and exemplar sentences. Coverage indicates the fraction of types attested in at least one exemplar. The lexicon associates an average of 12.8 LUs with a frame, and 66% of those LUs are attested for that frame. The average ambiguity of an LU is 1.2 frames (the 1322 ambiguous LUs have an average ambiguity of 2.4 frames).

TARGETS AND ARGUMENTS BY PART OF SPEECH					
	targets			**arguments**	
	count	*%*		*count*	*%*
Noun	5155	52	Noun	9439	55
Verb	2785	28	Preposition or		
Adjective	1411	14	complementizer	2553	15
Preposition	296	3	Adjective	1744	10
Adverb	103	1	Verb	1156	7
Number	63	1	Pronoun	736	4
Conjunction	8		Adverb	373	2
Article	3		Other	1047	6
	9824			17048	

Table 2: Breakdown of targets and arguments in the SemEval'07 training set in terms of part of speech. The target POS is based on the LU annotation for the frame instance. For arguments, this reflects the part of speech of the head word (estimated from automatic dependency parse); the percentage is out of all overt arguments.

possible output of our parser.

Table 3 provides a snapshot of the SemEval'07 data. We randomly selected four documents from the original SemEval training data to create a development set for tuning model hyperparameters. Notice that the test set contains more annotations per word, both in terms of frames and arguments. Moreover, there are many more out-of-lexicon frame, role, and LU types in the test set than in the training set. This inconsistency in the data results in poor recall scores for all models trained on the given data split, a problem we have not sought to address here.

Table 2 shows the breakdown of the targets and the arguments with respect to part of speech in the SemEval'07 training data. The statistics indicate that for both, nouns dominate the annotations, followed by verbs. However, unlike other corpora for semantic role labeling the FrameNet annotations encompass nearly all types of POS for the targets.

Preprocessing. We preprocess sentences in our dataset with a standard set of annotations: POS tags from MXPOST (Ratnaparkhi, 1996) and dependency parses from the MST parser (McDonald et al., 2005) since manual syntactic parses are not available for most of the FrameNet-annotated documents. We used WordNet (Fellbaum, 1998) for

FULL-TEXT ANNOTATIONS	*SemEval'07 data* train			dev			test		
Size	(*words*	*sentences*	*documents*)						
all	43.3K	1.7K	22	6.3K	251	4	2.8K	120	3
ANC (travel)	3.9K	154	2	.8K	32	1	1.3K	67	1
NTI (bureaucratic)	32.2K	1.2K	15	5.5K	219	3	1.5K	53	2
PropBank (news)	7.3K	325	5	0	0	0	0	0	0
Annotations	(*frames/word*	*overt arguments/word*)							
all	0.23	0.39		0.22	0.37		0.37	0.65	
ANC	0.22	0.38		0.15	0.29		0.37	0.60	
NTI	0.23	0.40		0.23	0.37		0.38	0.69	
PropBank	0.22	0.37							
Coverage of lexicon	(*% frames*	*% roles*	*% LUs*)						
all	64.1	27.4	21.0	34.0	10.2	7.3	29.3	7.7	4.9
ANC	26.4	7.4	4.8	8.9	2.0	1.1	17.5	3.9	2.3
NTI	52.4	21.1	14.9	31.5	9.2	6.7	19.0	5.0	3.0
PropBank	40.8	12.0	8.4						
Out-of-lexicon types	(*frames*	*roles*	*LUs*)						
all	14	69	71	2	4	2	39	99	189
ANC	12	39	41	0	0	2	26	63	123
NTI	6	32	33	2	4	0	19	45	70
PropBank	3	11	3						
Out-of-lexicon tokens	(*% frames*	*% roles*	*% LUs*)						
all	0.7	0.9	1.1	1.0	0.4	0.2	9.8	11.2	25.3
ANC	3.2	4.2	7.6	0.0	0.0	1.8	11.5	13.5	34.8
NTI	0.6	0.6	0.5	1.1	0.4	0.0	8.5	9.4	17.4
PropBank	0.3	0.4	0.2						

Table 3: Snapshot of the SemEval'07 annotated data. Our development set encompasses the following documents: `StephanopoulousCrimes` (from ANC), plus `Iran_Biological`, `NorthKorea_Introduction`, and `WMDNews_042106` (NTI). We use the standard test set, consisting of `IntroOfDublin` (ANC) and `chinaOverview` and `workAdvances` (NTI). Two ANC documents provided as part of the task were unannotated; we ignore them throughout.

lemmatization. Our models treat these pieces of information as observations. We also labeled each verb in the data as having ACTIVE or PASSIVE voice, using code from the SRL system described by Johansson and Nugues (2008).

2.3 Task and Evaluation

Automatic annotations of frame-semantic structure can be broken into three parts: (1) *targets*, the words or phrases that evoke frames; (2) the *frame type*, defined in the lexicon, evoked by each target; and (3) the *arguments*, or spans of words that serve to fill roles defined by each evoked frame. These correspond to the three subtasks in our parser, each described and evaluated in turn: target identification (§3), frame identification (§4, not unlike word-sense disambiguation), and argument identification (§5, not unlike semantic role labeling). Our parser is available for download at `http://www.ark.cs.cmu.edu/SEMAFOR`.

The standard evaluation script from the SemEval'07 shared task calculates precision, recall, and F_1-measure for frames and arguments; it also provides a score that gives partial credit for hypothesizing a frame related to the correct one. We present precision, recall, and F_1-measure microaveraged across the test documents, report *labels-only*

matching scores (spans must match exactly), and do not use named entity labels. More details can be found in Baker et al. (2007). For our experiments, statistical significance is measured using a reimplementation of Dan Bikel's randomized parsing evaluation comparator.[4]

2.4 Baseline

A strong baseline for frame-semantic parsing is the system presented by (Johansson and Nugues, 2007, hereafter J&N'07), the best system in the SemEval'07 shared task. That system is based on a collection of SVMs. For frame identification, they used an SVM classifier to disambiguate frames for known frame-evoking words. They used WordNet synsets to extend the vocabulary of frame-evoking words to cover unknown words, and then used a collection of separate SVM classifiers—one for each frame—to predict a single evoked frame for each occurrence of a word in the extended set.

J&N'07 modeled the argument identification problem by dividing it into two tasks: first, they classified candidate spans as to whether they were arguments or not; then they assigned roles to those that were identified as arguments. Both phases used SVMs. Thus, their formulation of the problem involves a multitude of classifiers—whereas ours uses two log-linear models, each with a single set of weights, to find a full frame-semantic parse.

3 Target Identification

Target identification is the problem of deciding which word tokens (or word token sequences) evoke frames in a given sentence. In other semantic role labeling schemes (e.g. PropBank), simple part-of-speech criteria typically distinguish predicates from non-predicates. But in frame semantics, verbs, nouns, adjectives, and even prepositions can evoke frames under certain conditions. One complication is that semantically-impoverished **support predicates** (such as *make* in *make a request*) do not evoke frames in the context of a frame-evoking, syntactially-dependent noun (*request*). Furthermore, only temporal, locative, and directional senses of prepositions evoke frames.

We found that, because the test set is more completely annotated—that is, it boasts far more frames per token than the training data (see Table 3)—learned models did not generalize well and achieved poor test recall. Instead, we followed J&N'07 in using a small set of rules to identify targets.

First, we created a master list of all the morphological variants of targets that appear in the exemplar sentences and the SemEval'07 training set. For a sentence in new data, we considered only those substrings as candidate targets, that appear in this master list. We also did not attempt to capture discontinuous frame targets: e.g. we treat *there would have been* as a single span even though the corresponding LU is *there be*.V.[5]

Next, we pruned the candidate target set by applying a series of rules identical to the ones described by (Johansson and Nugues, 2007, §3.1.1), with two exceptions. First, they identified locative, temporal, and directional prepositions using a dependency parser so as to retain them as valid LUs. In contrast, we pruned all types of prepositions because we found them to hurt our performance on the development set due to errors in syntactic parsing. In a second departure from their target extraction rules, we did not remove the candidate targets that had been tagged as support verbs for some other target.

[4] http://www.cis.upenn.edu/~dbikel/software.html#comparator

[5] There are 629 multiword LUs in the lexicon, and they correspond to 4.8% of the targets in the training set; among them are *screw up*.V, *shoot the breeze*.V, and *weapon of mass destruction*.N. In the SemEval'07 training data, there are just 99 discontinuous multiword targets (1% of all targets).

TARGET IDENTIFICATION	P	R	F_1
Our technique (§3)	**89.92**	**70.79**	**79.21**
Baseline: J&N'07	*87.87*	*67.11*	*76.10*

Table 4: Target identification results for our system and the baseline. Scores in bold denote significant improvements over the baseline ($p < 0.05$).

Note that we used a conservative white list which filters out targets whose morphological variants were not seen either in the lexicon or the training data. Therefore, our *full* parser loses the capability to predict frames for completely unseen LUs, despite the fact that our our powerful frame identification model (§4) can accurately label frames for new LUs.

Results. Table 4 shows results on target identification; our system gains 3 F_1 points over the baseline. This is statistically significant with $p < 0.01$. Our results are also significant in terms of precision ($p < 0.05$) and recall ($p < 0.01$). There are 85 distinct LUs for which the baseline fails to identify the correct target while our system succeeds. Considerable proportion of these units have more than one tokens (e.g. *chemical and biological weapon*.N, *ballistic missile*.N, etc.), which J&N'07 do not model. The baseline also does not label variants of *there be*.V, e.g. *there are* and *there has been*, which we correctly label as targets. Some examples of other single token LUs that the baseline fails to identify are names of months, LUs that belong to the ORIGIN frame (e.g. *iranian*.A) and directions, e.g., *north*.A or *north-south*.A.

4 Frame Identification

Given targets, the parser next identifies their frames.

4.1 Lexical units

FrameNet specifies a great deal of structural information both within and among frames. For frame identification we make use of frame-evoking **lexical units**, the (lemmatized and POS-tagged) words and phrases listed in the lexicon as referring to specific frames. For example, listed with the BRAGGING frame are 10 LUs, including *boast*.N, *boast*.V, *boastful*.A, *brag*.V, and *braggart*.N. Of course, due to polysemy and homonymy, the same LU may be associated with multiple frames; for example, *gobble*.V is listed under both the INGESTION and MAKE_NOISE frames. We thus term *gobble*.V an **ambiguous** LU (see Table 1).[6] All targets in the exemplar sentences, and most in our training and test data, correspond to known LUs (see Table 3).

To incorporate frame-evoking expressions found in the training data but not the lexicon—and to avoid the possibility of lemmatization errors—our frame identification model will incorporate, via a latent variable, features based directly on exemplar and training **targets** rather than LUs. Let $\mathcal{L}$ be the set of (unlemmatized and automatically POS-tagged) targets found in the exemplar sentences of the lexicon and/or the sentences in our training set. Let $\mathcal{L}_f \subseteq \mathcal{L}$ be the subset of these targets annotated as evoking a par-

[6] In our terminology an LU may be shared by multiple frames (LUs may be defined elswehere as frame-specific).

ticular frame f.[7] Let $\mathcal{L}^l$ and $\mathcal{L}^l_f$ denote the lemmatized versions of $\mathcal{L}$ and $\mathcal{L}_f$ respectively. Then, we write *boasted*.VBD $\in \mathcal{L}_{\textsc{Bragging}}$ and *boast*.VBD $\in \mathcal{L}^l_{\textsc{Bragging}}$ to indicate that this inflected verb *boasted* and its lemma *boast* have been seen to evoke the BRAGGING frame. Significantly, however, another target, such as *toot your own horn*, might be used in other data to evoke this frame. We thus face the additional hurdle of predicting frames for unknown words.

The SemEval annotators created 47 new frames not present in the lexicon, out of which 14 belonged to our training set. We considered these with the 795 frames in the lexicon when parsing new data. Predicting new frames is a challenge not yet attempted to our knowledge (including here). Note that the scoring metric (§2.3) gives partial credit for *related* frames (e.g., a more general frame from the lexicon).

4.2 Model

For a given sentence $\mathbf{x}$ with frame-evoking targets $\mathbf{t}$, let t_i denote the ith target (a word sequence).[8] Let t^l_i denote its lemma. We seek a list $\mathbf{f} = \langle f_1, \ldots, f_m \rangle$ of frames, one per target. In our model, the set of candidate frames for t_i is defined to include every frame f such that $t^l_i \in \mathcal{L}^l_f$—or if $t^l_i \notin \mathcal{L}^l$, then every known frame (the latter condition applies for 4.7% of the gold targets in the development set). In both cases, we let $\mathcal{F}_i$ be the set of candidate frames for the ith target in $\mathbf{x}$.

To allow frame identification for targets whose lemmas were seen in neither the exemplars nor the training data, our model includes an additional variable, ℓ_i. This variable ranges over the seen targets in $\mathcal{L}_{f_i}$, which can be thought of as **prototypes** for the expression of the frame. Importantly, frames are *predicted*, but prototypes are summed over via the latent variable. The prediction rule requires a probabilistic model over frames for a target:

$$f_i \leftarrow \operatorname*{argmax}_{f \in \mathcal{F}_i} \sum_{\ell \in \mathcal{L}_f} p(f, \ell \mid t_i, \mathbf{x}) \tag{1}$$

We adopt a conditional log-linear model: for $f \in \mathcal{F}_i$ and $\ell \in \mathcal{L}_f$, $p_{\boldsymbol{\theta}}(f, \ell \mid t_i, \mathbf{x}) =$

$$\frac{\exp \boldsymbol{\theta}^\top \mathbf{g}(f, \ell, t_i, \mathbf{x})}{\sum_{f' \in \mathcal{F}_i} \sum_{\ell' \in \mathcal{L}_{f'}} \exp \boldsymbol{\theta}^\top \mathbf{g}(f', \ell', t_i, \mathbf{x})} \tag{2}$$

where $\boldsymbol{\theta}$ are the model weights, and $\mathbf{g}$ is a vector-valued feature function. This discriminative formulation is very flexible, allowing for a variety of (possibly overlapping) features; e.g., a feature might relate a frame type to a prototype, represent a lexical-semantic relationship between a prototype and a target, or encode part of the syntax of the sentence.

Previous work has exploited WordNet for better coverage during frame identification (Johansson and Nugues, 2007; Burchardt et al., 2005, e.g., by expanding the set of targets using synsets), and others have sought to extend the lexicon itself (see §6). We differ in our use of a latent variable to incorporate lexical-semantic *features* in a discriminative model, relating known lexical units to unknown words that may evoke frames. Here we are able to take advantage of the large inventory of partially-annotated exemplar sentences.

Note that this model makes a strong independence assumption: each frame is predicted independently of all others in the document. In this way the model is similar to

[7] On average, there are 34 targets per frame in our dataset. The average frame ambiguity of each target in $\mathcal{L}$ is 1.17.

[8] Each t_i is a word sequence $\langle w_u, \ldots, w_v \rangle$, $1 \leq u \leq v \leq n$, though in principle targets can be noncontiguous

• the POS of the parent of the head word of t_i
• the set of syntactic dependencies of the head word[9] of t_i
• if the head word of t_i is a verb, then the set of dependency labels of its children
• the dependency label on the edge connecting the head of t_i and its parent
• the sequence of words in the prototype, $\mathbf{w}_\ell$
• the lemmatized sequence of words in the prototype
• the lemmatized sequence of words in the prototype and their part-of-speech tags $\boldsymbol{\pi}_\ell$
• WordNet relation[10] ρ holds between ℓ and t_i
• WordNet relation[10] ρ holds between ℓ and t_i, and the prototype is ℓ
• WordNet relation[10] ρ holds between ℓ and t_i, the POS tag sequence of ℓ is $\boldsymbol{\pi}_\ell$, and the POS tag sequence of t_i is $\boldsymbol{\pi}_t$

Table 5: Features used for frame identification. All also incorporate f, the frame being scored. $\ell = \langle \mathbf{w}_\ell, \boldsymbol{\pi}_\ell \rangle$ consists of the words and POS tags[11] of a target seen in an exemplar or training sentence as evoking f. There are a total of 662,020 binary features in our model.

J&N'07. However, ours is a single conditional model that shares features and weights across all targets, frames, and prototypes, whereas the approach of J&N'07 consists of many separately trained models. Moreover, our model is unique in that it uses a latent variable to smooth over frames for unknown or ambiguous LUs.

Frame identification features depend on the preprocessed sentence $\mathbf{x}$, the prototype ℓ and its WordNet lexical-semantic relationship with the target t_i, and of course the frame f. Our model instantiates 662,020 binary features.

4.3 Training

Given the training subset of the SemEval'07 data, which is of the form $\left\langle \langle \mathbf{x}^{(j)}, \mathbf{t}^{(j)}, \mathbf{f}^{(j)}, \mathcal{A}^{(j)} \rangle \right\rangle_{j=1}^{N}$ ($N = 1663$ is the number of sentences), we discriminatively train the frame identification model by maximizing the following log-likelihood:[12]

$$\max_{\boldsymbol{\theta}} \sum_{j=1}^{N} \sum_{i=1}^{m_j} \log \sum_{\ell \in \mathcal{L}_{f_i^{(j)}}} p_{\boldsymbol{\theta}}(f_i^{(j)}, \ell \mid t_i^{(j)}, \mathbf{x}^{(j)}) \tag{3}$$

Note that the training problem is non-convex because of the summed-out prototype latent variable ℓ for each frame. To calculate the objective function, we need to cope with a sum over frames and prototypes for each target (see Eq. 2), often an expensive operation. We locally optimize the function using a distributed implementation of L-BFGS. This is the most expensive model that we train: with 100 CPUs, training takes several hours. (Decoding takes only a few minutes on one CPU for the test set.)

[9] If the target is not a subtree in the parse, we consider the words that have parents outside the span, and apply three heuristic rules to select the head: 1) choose the first word if it is a verb; 2) choose the last word if the first word is an adjective; 3) if the target contains the word *of*, and the first word is a noun, we choose it. If none of these hold, choose the last word with an external parent to be the head.

[10] These are: IDENTICAL-WORD, SYNONYM, ANTONYM (including extended and indirect antonyms), HYPERNYM, HYPONYM, DERIVED FORM, MORPHOLOGICAL VARIANT (e.g., plural form), VERB GROUP, ENTAILMENT, ENTAILED-BY, SEE-ALSO, CAUSAL RELATION, and NO RELATION.

[11] POS tags are found automatically during preprocessing.

[12] We found no benefit on development data from using an L_2 regularizer (zero-mean Gaussian prior).

FRAME IDENTIFICATION (§4)	*targets*	**exact frame matching** *P*	*R*	F_1	**partial frame matching** *P*	*R*	F_1
Frame identification (oracle targets)	$*$	60.21	60.21	60.21	74.21	74.21	74.21
Frame identification (predicted targets)	auto §3	**69.75**	**54.91**	**61.44**	**77.51**	**61.03**	**68.29**
Baseline: J&N'07	*auto*	*66.22*	*50.57*	*57.34*	*73.86*	*56.41*	*63.97*

Table 6: Frame identification results. Precision, recall, and F_1 were evaluated under exact and partial frame matching; see §2.3. Bold indicates statistically significant results with respect to the baseline ($p < 0.05$).

4.4 Results

We evaluate the performance of our frame identification model given gold-standard targets and automatically identified targets (§3); see Table 6.

To compare the frame identification stage in isolation with that of J&N'07, we ran our frame identification model with the targets identified by their system as input. With partial matching, our model achieves a relative improvement of 0.6% F_1 over J&N'07 (though this is not significant).

While our frame identification model thus performs on par with the current state of the art for this task, it improves upon J&N's formulation of the problem because it requires only a single model, learns lexical-semantic features as part of that model rather than requiring a preprocessing step to expand the vocabulary of frame-evoking words, and is probabilistic, which can facilitate global reasoning.

For gold-standard targets, 210 out of 1058 lemmas were not present in the white list that we used for target identification (see §3). Our model correctly identifies the frames for 4 of these 210 lemmas. For 44 of these lemmas, the evaluation script assigns a score of 0.5 or more, suggesting that our model predicts a closely related frame. Finally, for 190 of the 210 lemmas, a positive score is assigned by the evaluation script. This suggests that the hidden variable model helps in identifying related (but rarely exact) frames for unseen targets, and explains why under exact—but not partial—frame matching, the F_1 score using automatic targets is commensurate with the score for oracle targets.[13]

For automatically identified targets, the F_1 score falls below 70 points because the model fails to predict frames for unseen lemmas. However, our model outperforms J&N'07 by 4 F_1 points. We measured statistical significance with respect the baseline for results with the partial frame matching criterion. The F_1 score of our model represents a significant improvement over the baseline ($p < 0.01$). The precision and recall measures are significant as well ($p < 0.05$ and $p < 0.01$, respectively). Note that the automatic target identification model leads to an increase in precision, at the expense of recall. This is because of the fact that the white list for target identification restricts the model to predict frames only for known LUs, leading to a more precise model.

5 Argument Identification

Given a sentence $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$, the set of targets $\mathbf{t} = \langle t_1, \ldots, t_m \rangle$, and a list of evoked frames $\mathbf{f} = \langle f_1, \ldots, f_m \rangle$ corresponding to each target, argument identification is the task

[13] J&N'07 did not report frame identification results for oracle targets; thus directly comparing the frame identification models is difficult. Considering only the predicted arguments for the frames they predicted correctly, we can estimate that their argument identification model given oracle targets and frames would have achieved 0.58 precision, 0.48 recall, and 0.53 F_1—though we caution that these are not directly comparable with our oracle results.

of choosing which of each f_i's roles are filled, and by which parts of $\mathbf{x}$. This task is most similar to the problem of semantic role labeling, but uses frame-specific labels that are richer than the PropBank annotations.

5.1 Model

Let $\mathcal{R}_{f_i} = \{r_1, \ldots, r_{|\mathcal{R}_{f_i}|}\}$ denote frame f_i's **roles** (named frame element types) observed in an exemplar sentence and/or our training set. A subset of each frame's roles are marked as **core** roles; these roles are conceptually and/or syntactically necessary for any given use of the frame, though they need not be overt in every sentence involving the frame. These are roughly analogous to the core arguments A0–A5 and AA in PropBank. Non-core roles—analogous to the various AMs in PropBank—loosely correspond to syntactic adjuncts, and carry broadly-applicable information such as the time, place, or purpose of an event. The lexicon imposes some additional structure on roles, including relations to other roles in the same or related frames, and semantic types with respect to a small ontology (marking, for instance, that the entity filling the protagonist role must be sentient for frames of cognition). Fig. 2 illustrates some of the structural elements comprising the frame lexicon by considering the CAUSE_TO_MAKE_NOISE frame.

We identify a set $\mathcal{S}$ of spans that are candidates for filling any role $r \in \mathcal{R}_{f_i}$. In principle, $\mathcal{S}$ could contain any subsequence of $\mathbf{x}$, but in this work we only consider the set of contiguous spans that (a) contain a single word or (b) comprise a valid subtree of a word and all its descendants in the dependency parse produced by the MST parser. This covers 81% of arguments in the development data. The empty span, denoted $\emptyset$, is also included in $\mathcal{S}$, since some roles are not explicitly filled; in the development data, the average number of roles an evoked frame defines is 6.7, but the average number of overt arguments is only 1.7.[14] In training, if a labeled argument is not a valid subtree of the dependency parse, we add its span to $\mathcal{S}$.

Let $\mathcal{A}_i$ denote the mapping of roles in $\mathcal{R}_{f_i}$ to spans in $\mathcal{S}$. Our model makes a prediction for each $\mathcal{A}_i(r_k)$ (for all roles $r_k \in \mathcal{R}_{f_i}$) using:

$$\mathcal{A}_i(r_k) \leftarrow \operatorname*{argmax}_{s \in \mathcal{S}} p(s \mid r_k, f_i, t_i, \mathbf{x}) \tag{4}$$

We use a conditional log-linear model over spans for each role of each evoked frame:

$$p_{\boldsymbol{\psi}}(\mathcal{A}_i(r_k) = s \mid f_i, t_i, \mathbf{x}) = \frac{\exp \boldsymbol{\psi}^\top \mathbf{h}(s, r_k, f_i, t_i, \mathbf{x})}{\sum_{s' \in \mathcal{S}} \exp \boldsymbol{\psi}^\top \mathbf{h}(s', r_k, f_i, t_i, \mathbf{x})} \tag{5}$$

Note that our model chooses the span for each role separately from the other roles and ignores all frames except the frame the role belongs to. Our model departs from the traditional SRL literature by modeling the argument identification problem in a single stage, rather than first classifying token spans as arguments and then labeling them.

[14] In the annotated data, each core role is filled with one of three types of *null instantiations* indicating how the role is conveyed implicitly. For instance, the imperative construction implicitly designates a role as filled by the addressee, and the corresponding filler is thus CNI (constructional null instantiation). In this work we do not distinguish different types of null instantiations.

[15] Quantized into groups: $(-\infty, -20]$, $[-19, -10]$, $[-9, -5]$, -4, -3, -2, -1, 0, 1, 2, 3, 4, $[5, 9]$, $[10, 19]$, $[20, \infty)$.

[16] We treat as a closed-class POS tag any Penn Treebank tag except for CD which does not start with V, N, A, or R.

Features with both null and non-null variants: These features come in two flavors: if the argument is null, then one version fires; if it is overt (non-null), then another version fires.	
● some word in t has lemma λ	● some word in t has POS π
◐ some word in t has lemma λ, and the sentence uses PASSIVE voice	◐ some word in t has lemma λ, and the sentence uses ACTIVE voice
◐ the head of t has subcategorization sequence $\boldsymbol{\tau} = \langle \tau_1, \tau_2, \ldots \rangle$	◐ some syntactic dependent of the head of t has dependency type τ
● the head of t has c syntactic dependents	● bias feature (always fires)
Span content features: apply to overt argument candidates.	
○ POS tag π occurs for some word in s	
○ the head word of s has POS π	● $\lvert s \rvert$, the number of words in the candidate argument[15]
○ the first word of s has POS π, provided $\lvert s \rvert > 0$	○ the head word of s has syntactic dependency type τ
○ the last word of s has POS π, provided $\lvert s \rvert > 0$	● the syntactic dependency type τ_{s_1} of the first word with respect to its head
● the first word of s: w_{s_1}, and its POS tag π_{s_1}, provided that π_{s_1} is a closed-class POS[16]	● τ_{s_2}, provided that $\lvert s \rvert \geq 2$
● w_{s_2} and its closed-class POS tag π_{s_2}, provided that $\lvert s \rvert \geq 2$	● $\tau_{s_{\lvert s \rvert}}$, provided that $\lvert s \rvert \geq 3$
● the last word of s: $w_{s_{\lvert s \rvert}}$, and its closed-class POS tag $\pi_{s_{\lvert s \rvert}}$, provided that $\lvert s \rvert \geq 3$	○ the first word of s has lemma λ, provided $\lvert s \rvert > 0$
	○ the head word of s has lemma λ
	○ the last word of s has lemma λ, provided $\lvert s \rvert > 0$
◐ lemma λ is realized in some word in s	◐ lemma λ is realized in some word in s, the voice denoted in the span, s's position with respect to t (BEFORE, AFTER, or OVERLAPPING)
◐ lemma λ is realized in some word in s, the voice denoted in the span (ACTIVE or PASSIVE)	
Syntactic features: apply to overt argument candidates.	
○ dependency path: sequence of labeled, directed edges from the head word of s to the head word of t	○ length of the dependency path[15]
Span context POS features: for overt candidates, up to 6 of these features will be active.	
○ a word with POS π occurs up to 3 words before the first word of s	○ a word with POS π occurs up to 3 words after the last word of s
Ordering features: apply to overt argument candidates.	
● the position of s with respect to to the span of t: BEFORE, AFTER, or OVERLAPPING (i.e. there is at least one word shared by s and t)	○ target-argument crossing: there is at least one word shared by s and t, at least one word in s that is not in t, and at least one word in t that is not in s
○ linear word distance between the nearest word of s and the nearest word of t, provided s and t do not overlap[15]	○ linear word distance between the middle word of s and the middle word of t, provided s and t do not overlap[15]

Table 7: Features used for argument identification. Instantiating the above (binary) features for our data yields 1,297,857 parameters.

A constraint implicit in our formulation restricts each role to have at most one overt argument, which is consistent with 96.5% of the role instances in the training data.

Out of the overt argument spans in the training data, 12% are duplicates, having been used by some other frame in the sentence. Our role-filling model, unlike a sentence-global argument detection-and-classification approach,[17] permits this sort of argument sharing among frames. The incidence of span overlap among frames is much higher; Fig. 1 illustrates a case with a high degree of overlap. Word tokens belong to an average of 1.6 argument spans, including the quarter of words that do not belong to any

[17] J&N'07, like us, identify arguments for each target.

argument.

Appending these local inference decisions together gives us the best mapping $\hat{\mathcal{A}}_t$ for target t. Features for our log-linear model (Eq. 5) depend on the preprocessed sentence $\mathbf{x}$; the target t; a role r of frame f; and a candidate argument span $s \in \mathcal{S}$.[18] For features using the head word of the target t or a candidate argument span s, we use the heuristic described in footnote 9 for selecting the head of non-subtree spans. Table 7 lists the feature templates used in our model. Every feature template has a version which does not take into account the role being filled (so as to incorporate overall biases). The ◐ symbol indicates that the feature template also has a variant which is conjoined with r, the name of the role being filled; and ● indicates that the feature template additionally has a variant which is conjoined with both r and f, the name of the frame.[19] The role name–only variants provide for smoothing over frames for common types of roles such as Time and Place; see Matsubayashi et al. (2009) for a detailed analysis of the effects of using role features at varying levels of granularity.

5.2 Training

We train the argument identification model by:

$$\max_{\boldsymbol{\psi}} \sum_{j=1}^{N} \sum_{i=1}^{m_j} \sum_{k=1}^{|\mathcal{R}_{f_i^{(j)}}|} \log p_{\boldsymbol{\psi}}(\mathcal{A}_i^{(j)}(r_k) \mid f_i^{(j)}, t_i^{(j)}, \mathbf{x}^{(j)}) \tag{6}$$

This objective function is concave, and we globally optimize it using stochastic gradient ascent (Bottou, 2004). We train this model until the argument identification F_1 score stops increasing on the development data. Best results on this dataset were obtained with a batch size of 2 and 23 passes through the development data.

Algorithm 1 Joint decoding of frame f_i's arguments. $\mathsf{top_k}(\mathcal{S}, p_{\boldsymbol{\psi}}, r_j)$ extracts the k most probable spans from $\mathcal{S}$, under $p_{\boldsymbol{\psi}}$, for role r_j. $\mathsf{extend}(D^{0:(j-1)}, \mathcal{S}')$ extends each span vector in $D^{0:(j-1)}$ with the most probable non-overlapping span from $\mathcal{S}'$, resulting in k best extensions overall.

Input: $\mathsf{k} > 0$, $\mathcal{R}_{f_i}$, $\mathcal{S}$, the distribution $p_{\boldsymbol{\psi}}$ from Eq. 5 for each role $r_j \in \mathcal{R}_{f_i}$

Output: $\hat{\mathcal{A}}_i$, a high-scoring mapping of roles of f_i to spans with no token overlap among the spans

Calculate $\mathcal{A}_i$ according to Eq. 4
$\forall r \in \mathcal{R}_{f_i}$ such that $\mathcal{A}_i(r) = \emptyset$, let $\hat{\mathcal{A}}_i(r) \leftarrow \emptyset$
$\mathcal{R}_{f_i}^{+} \leftarrow \{r : r \in \mathcal{R}_{f_i}, \mathcal{A}_i(r) \neq \emptyset\}$
$\mathsf{n} \leftarrow |\mathcal{R}_{f_i}^{+}|$
Arbitrarily order $\mathcal{R}_{f_i}^{+}$ as $\{r_1, r_2, \ldots r_{\mathsf{n}}\}$
Let $D^{0:j} = \langle D_1^{0:j}, \ldots, D_{\mathsf{k}}^{0:j} \rangle$ refer to the k-best list of vectors of compatible filler spans for roles r_1 through r_j
Initialize $D^{0:0}$ to be empty
for $j = 1$ to n **do**
    $D^{0:j} \leftarrow \mathsf{extend}(D^{0:(j-1)}, \mathsf{top_k}(\mathcal{S}, p_{\boldsymbol{\psi}}, r_j))$
end for
$\forall j \in \{1, \ldots, \mathsf{n}\}, \hat{\mathcal{A}}_i(r_j) \leftarrow D_1^{0:\mathsf{n}}[j]$
return $\hat{\mathcal{A}}_i$

[18] In this section we use t, f, and r without subscripts since the features only consider a single role of a single target's frame.

[19] i.e., the ● symbol subsumes ◐, which in turn subsumes ○.

ARGUMENT IDENTIFICATION					exact frame matching					
	targets	*frames*	*spans*	*decoding*	*P*	*R*	F_1			
Argument identification	$*$	$*$	$*$	naïve	86.61	75.11	80.45			
(oracle spans)	$*$	$*$	$*$	beam §5.3	88.29	74.77	80.97			
Argument identification	$*$	$*$	model §5	naïve	77.43	60.76	68.09	**partial frame matching**		
(full)	$*$	$*$	model §5	beam §5.3	78.71	60.57	68.46	*P*	*R*	F_1
Parsing (oracle targets)	$*$	model §4	model §5	beam §5.3	49.68	42.82	46.00	57.85	49.86	53.56
Parsing (full)	auto §3	model §4	model §5	beam §5.3	**58.08**	**38.76**	**46.49**	**62.76**	**41.89**	**50.24**
Baseline: J&N'07	*auto*	*model*	*model*	*N/A*	*51.59*	*35.44*	*42.01*	*56.01*	*38.48*	*45.62*

Table 8: Argument identification results. $*$ indicates that gold-standard labels were used for a given pipeline stage. For full parsing, bolded scores indicate significant improvements relative to the baseline ($p < 0.05$).

5.3 Approximate Joint Decoding

Naïve prediction of roles using Eq. 4 may result in overlap among arguments filling different roles of a frame, since the argument identification model fills each role independently of the others. We want to enforce the constraint that two roles of a single frame cannot be filled by overlapping spans.[20] Toutanova et al. (2005) presented a dynamic programming algorithm to prevent overlapping arguments for semantic role labeling; however, their approach used an orthogonal view to the argument identification stage, wherein they labeled phrase-structure tree constituents with semantic roles. This view helped them to adopt a dynamic programming approach, which does not suit our model because we find best possible argument spans for a particular role.

To eliminate illegal overlap, we adopt the beam search technique detailed in Algorithm 1. The algorithm produces a set of k-best hypotheses for a frame instance's full set of role-span pairs, but uses an approximation in order to avoid scoring an exponential number of hypotheses. After determining which roles are most likely not explicitly filled, it considers each of the other roles in turn: in each iteration, hypotheses incorporating a subset of roles are extended with high-scoring spans for the next role, always maintaining k alternatives. We set $\mathsf{k} = 10000$.

5.4 Results

Performance of the argument identification model is presented in Table 8. The table shows how performance varies given different types of perfect input: correct targets, correct frames, and the set of correct spans; correct targets and frames, with the heuristically-constructed set of candidate spans; correct targets only, with model frames; and ultimately, no oracle input (the full frame parsing scenario).

The first four rows of results isolate the argument identification task from the frame identification task. Given gold targets and frames and an oracle set of argument spans, our local model achieves about 87% precision and 75% recall. Beam search decoding to eliminate illegal argument assignments within a frame (§5.3) further improves precision by about 1.6%, with negligible harm to recall. Note that 96.5% recall is possible under the constraint that roles are not multiply-filled (§5.1); there is thus considerable room for improvement with this constraint in place. Joint prediction of each frame's arguments is

[20] On rare occasions a frame annotation may include a *secondary frame element layer*, allowing arguments to be shared among multiple roles in the frame; see Ruppenhofer et al. (2006) for details. The evaluation for this task only considers the primary layer, which is guaranteed to have disjoint arguments.

# gold args %	# predicted args %				
	0	1	2	3	4
0	**25** 2.6	9 .9	4 .4	0	0
1	82 8.5	**248** 25.8	28 2.9	2 .2	0
2	36 3.7	190 19.8	**180** 18.7	10 1.0	0
3	7 .7	38 4.0	54 5.6	**21** 2.2	1 .1
4	3 .3	9 .9	1 .1	7 .7	**2** .2
5	0	1 .1	1 .1	2 .2	0

Table 9: Number of overt gold vs. predicted arguments for test set (given gold targets and frames, no beam decoding). Mass is concentrated on and to the left of the diagonal, indicating that the model is conservative about predicting arguments.

worth exploring to capture correlations not encoded in our local models or joint decoding scheme.

The 15-point drop in recall when the heuristically-built candidate argument set replaces the set of true argument spans is unsurprising: an estimated 19% of correct arguments are excluded because they are neither single words nor complete subtrees (see §5.1).[21] Qualitatively, the problem of candidate span recall seems to be largely due to syntactic parse errors.[22] Still, the 10-point decrease in precision when using the syntactic parse to determine candidate spans suggests that the model has trouble discriminating between good and bad arguments, and that additional feature engineering or jointly decoding arguments of a sentence's frames may be beneficial in this regard.

The fifth and sixth rows show the effect of automatic frame identification on overall frame parsing performance. There is a 22% decrease in F_1 (18% when partial credit is given for related frames), suggesting that improved frame identification or joint prediction of frames and arguments is likely to have a sizeable impact on overall performance.

The final two rows of the table compare our full model (target, frame, and argument identification) with the baseline, showing significant improvement of more than 4.4 F_1 points for both exact and partial frame matching. As with frame identification, we compared the argument identification stage with that of J&N'07 in isolation, using the automatically identified targets and frames from the latter as input to our model. With partial frame matching, this gave us an F_1 score of 48.1% on the test set—significantly better ($p < 0.05$) than 45.6%, the full parsing result from J&N'07 (last row in Table 8). This indicates that our argument identification model—which uses a single discriminative model with a large number of features for role filling (rather than argument labeling)—is more powerful than the previous state of the art.

6 Related work

Since Gildea and Jurafsky (2002) pioneered statistical semantic role labeling, a great deal of computational work has investigated predicate-argument structures for semantics.

[21] Using all constituents from the 10-best syntactic parses would improve oracle recall of spans in the development set by just a couple of percentage points, at the computational cost of a larger pool of candidate arguments per role.

[22] Note that, because of our labels-only evaluation scheme (§2.3), arguments missing a word or containing an extra word receive no credit. In fact, of the frame roles correctly predicted as having an overt span, the correct span was predicted 66% of the time, while 10% of the time the predicted starting and ending boundaries of the span were off by a total of 1 or 2 words.

# gold arg lengths %	# predicted arg lengths % 0	1	2	3	4	5–9	10–19	20+
0		60 3.3	26 1.4	22 1.2	8 0.4	19 1.1	5 0.3	1 0.1
1	316 17.5	**514 47.9**	10 0.9	15 1.4	6 0.6	12 1.1	7 0.7	5 0.5
2	94 5.2	41 3.8	**71 6.6**	1 0.1	2 0.2	10 0.9	7 0.7	0
3	62 3.4	15 1.4	1 0.1	**42 3.9**	1 0.1	12 1.1	4 0.4	2 0.2
4	28 1.6	13 1.2	2 0.2	6 0.6	**17 1.6**	4 0.4	4 0.4	1 0.1
5–9	66 3.7	28 2.6	8 0.7	17 1.6	7 0.7	**84 7.8**	10 0.9	3 0.3
10–19	16 0.9	11 1.0	2 0.2	5 0.5	1 0.1	11 1.0	**34 3.2**	2 0.2
20+	9 0.5	3 0.3	0	2 0.2	0	6 0.6	5 0.5	**8 0.7**

Table 10: Number of words in spans filling the same role for gold and predicted arguments (on test set, given gold targets and frames, no beam decoding). Again, the model is somewhat conservative, predicting more erroneous short spans than erroneous long spans—but if a span is predicted for the correct role, it will more likely have the right (quantized) length than fall into any other single length range. The most frequent mistake is predicting null when there should be a one-word argument. 0 refers to the null span. The percentages in the 0 row and column are not comparable with the percentages in the rest of the table.

Briefly, we highlight some relevant work, particularly research that has made use of FrameNet. (Note that much related research has focused on PropBank (Kingsbury and Palmer, 2002), a set of shallow predicate-argument annotations for *Wall Street Journal* articles from the Penn Treebank (Marcus et al., 1993); a recent issue of *CL* (Màrquez et al., 2008) was devoted to the subject.)

Most work on frame-semantic role labeling has made use of the exemplar sentences in the FrameNet corpus (see §2.1), each of which is annotated for a single frame and its arguments. On the probabilistic modeling front, Gildea and Jurafsky (2002) presented a discriminative model for arguments given the frame; Thompson et al. (2003) used a generative model for both the frame and its arguments; and Fleischman et al. (2003) first used maximum entropy models to find and label arguments given the frame. Shi and Mihalcea (2004) developed a rule-based system to predict frames and their arguments in text, and Erk and Padó (2006) introduced the Shalmaneser tool, which employs Naïve Bayes classifiers to do the same. Other FrameNet SRL systems (Giuglea and Moschitti, 2006, for instance) have used SVMs. Most of this work was done on an older, smaller version of FrameNet.

Recent work on frame-semantic *parsing*—in which sentences may contain multiple frames to be recognized along with their arguments—has used the SemEval'07 data (Baker et al., 2007). The LTH system of Johansson and Nugues (2007), our baseline (§2.4), performed the best in the SemEval'07 task. Matsubayashi et al. (2009) trained a log-linear model on the SemEval'07 data to evaluate argument identification features exploiting various types of taxonomic relations to generalize over roles. Another line of work has sought to extend the coverage of FrameNet by exploiting VerbNet, WordNet, and Wikipedia (Shi and Mihalcea, 2005; Giuglea and Moschitti, 2006; Pennacchiotti et al., 2008; Tonelli and Giuliano, 2009), and projecting entries and annotations within and across languages (Boas, 2002; Fung and Chen, 2004; Padó and Lapata, 2005; Fürstenau and Lapata, 2009). Others have applied frame-semantic structures to question answering, paraphrase/entailment recognition, and information extraction (Narayanan and Harabagiu, 2004; Shen and Lapata, 2007; Padó and Erk, 2005; Burchardt, 2006; Bur-

chardt et al., 2009; Moschitti et al., 2003; Surdeanu et al., 2003). Wu and Fung (2009) explored PropBank SRL for machine translation. Recent work using PropBank has also explored online discriminative training (Pradhan et al., 2004), joint inference via reranking (Toutanova et al., 2005), integer linear programming (Punyakanok et al., 2004), and other algorithmic techniques.[23]

7 Conclusion

We have provided a supervised model for rich frame-semantic parsing, based on a combination of knowledge from FrameNet, two probabilistic models trained on SemEval'07 data, and expedient heuristics. Our system achieves improvements over the state of the art at each stage of processing and collectively, and is amenable to future extension. We have released a software package that implements the methods described in this report and is publicly available for use.

Acknowledgments

We thank Collin Baker, Katrin Erk, Richard Johansson, and Nils Reiter for software, data, evaluation scripts, and methodological details. We thank the reviewers, Alan Black, Ric Crabbe, Michael Ellsworth, Rebecca Hwa, Dan Klein, Russell Lee-Goldman, Dan Roth, Josef Ruppenhofer, and members of the ARK group for helpful comments. This work was supported by DARPA grant NBCH-1080004, NSF grant IIS-0836431, and computational resources provided by Yahoo.

References

Baker, C., Ellsworth, M., and Erk, K. (2007). SemEval-2007 Task 19: frame semantic structure extraction. In *Proc. of SemEval*. [3, 7, 17]

Boas, H. C. (2002). Bilingual FrameNet dictionaries for machine translation. In *Proc. of LREC*. [17]

Bottou, L. (2004). Stochastic learning. In *Advanced Lectures on Machine Learning*. Springer-Verlag. [14]

Burchardt, A. (2006). Approaching textual entailment with LFG and FrameNet frames. In *Proc. of the Second PASCAL RTE Challenge Workshop*. [-]

Burchardt, A., Erk, K., and Frank, A. (2005). A WordNet detour to FrameNet. In Fisseni, B., Schmitz, H., Schröder, B., and Wagner, P., editors, *Sprachtechnologie, mobile Kommunikation und linguistische Resourcen*, volume 8. Peter Lang. [9]

Burchardt, A., Pennacchiotti, M., Thater, S., and Pinkal, M. (2009). Assessing the impact of frame semantics on textual entailment. *Natural Language Engineering*, 15. [-]

Das, D., Schneider, N., Chen, D., and Smith, N. A. (2010). Probabilistic frame-semantic parsing. In *Proc. of NAACL-HLT*. [2]

Erk, K. and Padó, S. (2006). Shalmaneser - a toolchain for shallow semantic parsing. In *Proc. of LREC*. [17]

[23] The interested reader is encouraged to consult a recent special issue of *CL* (Màrquez et al., 2008).

Fellbaum, C., editor (1998). *WordNet: an electronic lexical database*. MIT Press, Cambridge, MA. [5]

Fillmore, C. J. (1982). Frame semantics. In *Linguistics in the Morning Calm*, pages 111–137. Hanshin Publishing Co., Seoul, South Korea. [3]

Fillmore, C. J., Johnson, C. R., and Petruck, M. R. (2003). Background to FrameNet. *International Journal of Lexicography*, 16(3). [3]

Fleischman, M., Kwon, N., and Hovy, E. (2003). Maximum entropy models for FrameNet classification. In *Proc. of EMNLP*. [17]

Fung, P. and Chen, B. (2004). BiFrameNet: bilingual frame semantics resource construction by cross-lingual induction. In *Proc. of COLING*. [17]

Fürstenau, H. and Lapata, M. (2009). Semi-supervised semantic role labeling. In *Proc. of EACL*. [17]

Gildea, D. and Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational Linguistics*, 28(3). [16, 17]

Giuglea, A. and Moschitti, A. (2006). Shallow semantic parsing based on FrameNet, VerbNet and PropBank. In *Proc. of ECAI 2006*. [17]

Johansson, R. and Nugues, P. (2007). LTH: semantic structure extraction using nonprojective dependency trees. In *Proc. of SemEval*. [7, 9, 17]

Johansson, R. and Nugues, P. (2008). Dependency-based semantic role labeling of PropBank. In *Proc. of EMNLP*. [6]

Kingsbury, P. and Palmer, M. (2002). From TreeBank to PropBank. In *Proc. of LREC*. [17]

Marcus, M. P., Marcinkiewicz, M. A., and Santorini, B. (1993). Building a large annotated corpus of English: the Penn Treebank. *Computational Linguistics*, 19(2). [17]

Màrquez, L., Carreras, X., Litkowski, K. C., and Stevenson, S. (2008). Semantic role labeling: an introduction to the special issue. *Computational Linguistics*, 34(2). [17, 18]

Matsubayashi, Y., Okazaki, N., and Tsujii, J. (2009). A comparative study on generalization of semantic roles in FrameNet. In *Proc. of ACL-IJCNLP*. [14, 17]

McDonald, R., Crammer, K., and Pereira, F. (2005). Online large-margin training of dependency parsers. In *Proc. of ACL*. [5]

Moschitti, A., Morărescu, P., and Harabagiu, S. M. (2003). Open-domain information extraction via automatic semantic labeling. In *Proc. of FLAIRS*. [17]

Narayanan, S. and Harabagiu, S. (2004). Question answering based on semantic structures. In *Proc. of COLING*. [17]

Padó, S. and Erk, K. (2005). To cause or not to cause: cross-lingual semantic matching for paraphrase modelling. In *Proc. of the Cross-Language Knowledge Induction Workshop*. [17]

Padó, S. and Lapata, M. (2005). Cross-linguistic projection of role-semantic information. In *Proc. of HLT-EMNLP*. [17]

Pennacchiotti, M., Cao, D. D., Basili, R., Croce, D., and Roth, M. (2008). Automatic induction of FrameNet lexical units. In *Proc. of EMNLP*. [17]

Pradhan, S. S., Ward, W. H., Hacioglu, K., Martin, J. H., and Jurafsky, D. (2004). Shallow semantic parsing using Support Vector Machines. In *Proc. of HLT-NAACL*. [18]

Punyakanok, V., Roth, D., W.-T. Yih, and Zimak, D. (2004). Semantic role labeling via integer linear programming inference. In *Proc. of COLING*. [18]

Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In *Proc. of EMNLP*. [5]

Ruppenhofer, J., Ellsworth, M., Petruck, M. R. L., Johnson, C. R., and Scheffczyk, J. (2006). FrameNet II: extended theory and practice. [15]

Shen, D. and Lapata, M. (2007). Using semantic roles to improve question answering. In *Proc. of EMNLP-CoNLL*. [17]

Shi, L. and Mihalcea, R. (2004). An algorithm for open text semantic parsing. In *Proc. of Workshop on Robust Methods in Analysis of Natural Language Data*. [17]

Shi, L. and Mihalcea, R. (2005). Putting pieces together: combining FrameNet, VerbNet and WordNet for robust semantic parsing. In *Computational Linguistics and Intelligent Text Processing: Proc. of CICLing 2005*. Springer-Verlag. [17]

Surdeanu, M., Harabagiu, S., Williams, J., and Aarseth, P. (2003). Using predicate-argument structures for information extraction. In *Proc. of ACL*. [17]

Thompson, C. A., Levy, R., and Manning, C. D. (2003). A generative model for semantic role labeling. In *Proc. of ECML*. [17]

Tonelli, S. and Giuliano, C. (2009). Wikipedia as frame information repository. In *Proc. of EMNLP*. [17]

Toutanova, K., Haghighi, A., and Manning, C. (2005). Joint learning improves semantic role labeling. In *Proc. of ACL*. [15, 18]

Wu, D. and Fung, P. (2009). Semantic roles for SMT: a hybrid two-pass model. In *Proc. of HLT-NAACL*. [18]