[NS provisional title:]Leveraging Additional Resources for Frame-Semantic Role Labeling

Abstract

The high cost of semantic structure annotation is a major obstacle to automating semantic analysis with broad coverage. The fully annotated datasets that exist are often small, hindering the robustness of models trained on them. However, low-resource tasks may benefit from exploiting *partially* annotated data, as well as data with *different* (but related) forms of annotation, for additional training data or features. This paper considers the argument identification and classification subtask of frame-semantic parsing, which to date has relied exclusively upon full-text annotations in the FrameNet resource.[NS is this true? I think Dipanjan used exemplars for the latent variable frame ID model, but he hasn't used them at all for arg ID, right?] We augment supervised learning with additional "indirect" training data and features so as to leverage additional resources internal and external to FrameNet (e.g., PropBank). [NS result]

1 Introduction

[NS sparseness is a challenge for many computational semantics tasks]

Frame-semantic parsing (Das et al., 2014) is a case in point. This is the task of automating the rich linguistic structure analyses of the FrameNet lexicon and corpus (Baker et al., 1998).[1] FrameNet represents kinds of events and other scenarios with an inventory of **frames** ([NS examples]). Each frame is associated with lexical **predicates** (verbs, nouns, adjectives, and adverbs) capable of evoking the scenario, and a set of **roles** (or **frame elements**) called to mind in order to understand the scenario. These roles may be implicit, but are frequently realized linguistically in the same sentence as the predicate. Given a sentence, frame-semantic parsing is the task of mapping tokens in the sentences to evoked frames, and for each evoked frame, finding and labeling its **argument** phrases with roles. An example appears in figure 1; it will be explained in detail in §2.1.

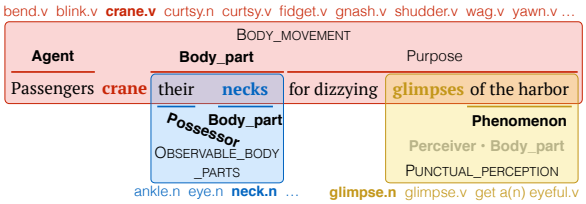

Figure 1: Example sentence from FrameNet full-text annotation. Three frames and their arguments are shown: BODY_MOVEMENT is evoked by *crane*, OBSERVABLE_BODY_PARTS by *necks*, and PUNCTUAL_PERCEPTION by *glimpse*. (Further, *harbor* is annotated as evoking the LOCALE_BY_USE frame and doubles as its sole argument.) Horizontal lines representing argument spans are labeled with role names.

FrameNet 1.5 defines a structured hierarchy of over 1,000 frames associated with [NS #] English lexical predicates, and also provides annotations for [NS # targets annotated total] attestations of these frames/predicates in corpora, annotated in context with their arguments. In FrameNet 1.5, a rather small number of sentences—[NS #], comprising [NS #] words—are provided with **full-text** annotations, i.e. the sentence has been analyzed for all available frames. But a full [NS #]% of sentences in FrameNet—the lexicographic **exemplars**—are annotated for only one frame per sentence, and have thus far not been exploited successfully for frame-semantic parsing. Here, we seek to leverage these exemplar sentences as well as the (type-level) hierarchical structure of the FrameNet lexicon.

In this paper, we address the **argument identification** subtask of finding and labeling arguments given a predicate in context and the frame it evokes.

[1] http://framenet.icsi.berkeley.edu/

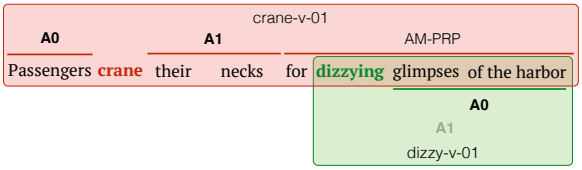

Figure 2: Ideal PropBank annotations for verbs. Though PB uses lexical frames rather than deep frames, there are clear similarities to the FrameNet annotations in figure 1.

This is a form of semantic role labeling (SRL). Notably, another resource, **PropBank** (Kingsbury and Palmer, 2002), has been widely used for SRL (Palmer et al., 2010). PropBank annotations capture shallower lexical frames and arguments; additionally, PropBank provides [$_S^{NS}$ millions?] of words of fully annotated English sentences (annotation is much less expensive, but also potentially less valuable, because of the shallower representation). Despite a number of differences in the representations and annotation conventions, for many predicates FrameNet and PropBank are really quite similar: figure 1 and figure 2 show this for the verb *crane*. To get the best of both worlds, we aim to tap into PropBank's vast resources as indirect token-level supervision for FrameNet-style analysis. We hypothesize that PropBank analyses can serve as a weak signal for the FrameNet SRL task, either by heuristically transforming PropBank annotations into FrameNet annotations to augment the training data, or by preprocessing sentences with a PropBank SRL system to obtain new features for FrameNet argument identification.

Our experiments expand the *training data* and/or the *feature space* of supervised argument identification in order to integrate evidence from all of these sources into SEMAFOR (Das et al., 2014), the leading open-source frame-semantic parser for English.[2] The results show that some of these sources of evidence succeed at boosting argument identification performance.[$_S^{NS}$ SOTA (without constraints)?]

2 Resources

[$_S^{NS}$ incl. data analysis of differences from full-text]

[$_S^{NS}$ w/in each mention: genre/overall vocabulary; coverage and distributions of predicates, FE labels; oracle coverage of FT test]

[2] http://www.ark.cs.cmu.edu/SEMAFOR/

2.1 Full-text Annotations

Contemporary frame-semantic parsers are trained and evaluated on the **full-text (FT)**[3] portion of the FrameNet corpus. This consists of documents for which annotators made an effort to assign frames and arguments to as many words as possible. Figure 1 gives an example sentence from the FT portion of the corpus. It has 4 frame annotations: [$_S^{NS}$ explain the figure].

In some cases, FT annotation involved creating a new frame or adding a new lexical unit to an existing frame. In other cases, words that in principle should be considered to evoke a frame were left unannotated because they did not match any existing lexical units. This was the case for *passengers* and *dizzying* in figure 1.

Genres, sizes

Annotation density: (proportion of tokens evoking a frame. breakdown by POS?)

2.2 Exemplars

[$_S^{NS}$ how different from FT]

2.3 PropBank

[$_S^{NS}$ how different from FN]

2.4 SemLink

[$_S^{NS}$ limitations!]

2.5 Illinois SRL system

3 Learning from multiple domains and representations

We experiment with several techniques for modifying the model-fitting portion of the argument identification model's local learning objective (training data, features). All of our experiments use the same form of regularization, condition on the same frame predictions[$_S^{NS}$ not oracle, right? these are SEMAFOR's current frame ID model, so not state of the art?] and syntactic preprocessing[$_S^{NS}$ does this match Dipanjan's latest experiments?], and use beam search with [$_S^{NS}$ hyperparam value] for joint decoding of arguments.[4]

[$_S^{NS}$ Domain adaptation/multitask learning techniques]

[3] Though these were *annotated* and the document level, and train/dev/test splits are by document, the frame-semantic parsing is currently restricted to the sentence level.

[4] [$_S^{NS}$ recent work has improved upon global decoding techniques; we expect such improvements to be complementary to the gains due to the local model reported here]

3.1 Augmenting the Training Data

3.2 Frustratingly Easy

3.3 2-stage

3.4 Type-level hierarchy features

4 Experiments

[$^{NS}_{S}$ tuning regularizer for all experiments]

4.1 Baseline

As a baseline, we compare to SEMAFOR (Das et al., 2014). We have made several modifications to SEMAFOR's training that do not affect performance, but do speed up experiments:

- We use squared structured hinge loss (defined below) instead of multiclass logistic regression. Using hinge loss, there is no longer a need to calculate a partition function. Gradients, and hence parameters, are sparser than in logistic regression, allowing us to use a sparse vector implementation.
- We use the online optimization method AdaDelta (Zeiler, 2012) with minibatches[$^{NS}_{S}$ minibatch size?], instead of the batch method L-BFGS (Liu and Nocedal, 1989).
- We use elastic net ($\ell_1 + \ell_2$) regularization. Adding ℓ_1 also has the effect of keeping the parameter vector sparse.[$^{NS}_{S}$ what hyperparameter(s)? are ℓ_1 and ℓ_2 tuned separately?]

We use these changes for all systems, including the baseline. While performance is not significantly affected[$^{S}_{T}$ we checked this, right?][$^{NS}_{S}$ meghana should have P/R/F and runtime numbers], these changes enabled us to run more experiments with the larger exemplar dataset and expanded feature space.

The details of squared hinge loss are as follows. We use a linearly parametrized model [$^{S}_{T}$ this should probably be talked about earlier]:

$$\text{score}_{\mathbf{w}}(y \mid x) = \mathbf{w}^\top \mathbf{f}(x, y) \tag{1}$$

Let $(x^{(i)}, y^{(i)})$ be the i^{th} training example. Then the structured hinge loss on the i^{th} example is given by:

$$Hinge_{\mathbf{w}}(i) = \max_y \{\mathbf{w}^\top \mathbf{f}(x^{(i)}, y) + \text{cost}(y, y^{(i)})\} - \mathbf{w}^\top \mathbf{f}(x^{(i)}, y^{(i)})$$

and squared hinge loss is:

$$SquaredHinge_{\mathbf{w}}(i) = Hinge_{\mathbf{w}}(i)^2. \tag{2}$$

In our experiments, we use $\text{cost}(y, y^{(i)}) = \mathbf{1}\{y \neq y^{(i)}\}$ [$^{S}_{T}$ define][5].

During training, each frame element is treated as an independent training instance. But at test time, all frame elements for a given frame are decoded at once, and beam search is used to ensure that no roles[$^{NS}_{S}$ arguments?] overlap.[$^{NS}_{S}$ does beam search require normalizing to probabilities?]

[$^{NS}_{S}$ what preprocessing? MSTParser?]

[$^{NS}_{S}$ can mention recall-oriented experiments in a footnote]

[$^{NS}_{S}$ same FN 1.5 splits as Dipanjan]

[$^{NS}_{S}$ beam search decoding, not fancy hard constraints]

[$^{NS}_{S}$ preprocessing issues: removing duplicate sentences, merging adjacent split args in exemplars, OntoNotes PropBank preprocessing (NLTK), token-level SemLink details (such as filtering out sentences without mappable annotations; copy from WS paper)]

4.2 Evaluation

[$^{NS}_{S}$ main eval: FT test; new eval: exemplars]

4.3 Results

[$^{NS}_{S}$ results table without hierarchy features]

[$^{NS}_{S}$ hierarchy features: which ones work best (decided on baseline), how do they improve best result so far]

[$^{NS}_{S}$ Sam's curves on per-FE F_1]

[$^{NS}_{S}$ comparison to prior work (baseline, best result). args+frames score vs. args only]

[$^{NS}_{S}$ discussion throughout]

5 Related Work

[$^{NS}_{S}$ Dipanjan's other papers; mention other PB SRL work?; anything using SemLink or combining resources for SRL?]

[$^{NS}_{S}$ multitask learning?]

6 Conclusion

[$^{NS}_{S}$ overall findings]

[$^{NS}_{S}$ future work: testing ground for improvements to PB and SemLink; automatic mappings between resources]

[5] We experimented with recall-oriented training, where errors of omission are assigned a higher cost, but found that while recall increased, overall F_1 went down[$^{NS}_{S}$ or: failed to improve?].

Additional Resource	Training Configuration (Features)	Full-Text			Exemplars		
		P	R	F_1	P	R	F_1
(Baseline)	FT (Basic)	66.03	53.79	59.29	64.90	33.60	44.27
FN Hierarchy	FT (siblings) FT (siblings+parents)						
Exemplars	Exemplars $\xrightarrow{\text{guide}}$ FT FT+Exemplars (Basic) FT+Exemplars (EasyAdapt)						
SemLink	SemLink $\xrightarrow{\text{guide}}$ FT FT+SemLink						
PB-SRL	FT (PB-SRL)						

Table 1: Results on two test sets: Baseline vs. individual other resources. Precision, recall, and F_1 are given as percentages.

FT+Exemplars (Hier: siblings+parents)
FT+Exemplars (PB-SRL)
FT+Exemplars (PB-SRL, Hier: siblings+parents)

Table 2: Combining best techniques across resources [$^{\text{NS}}_{\text{S}}$ TODO]

	Test on FN			Test on Exemplars		
	P	R	F_1	P	R	F_1
Semafor baseline (trained on FN)	0.6603	0.5379	0.5929			
baseline trained on combined	0.66061	0.58234	0.61901	0.75443	0.65107	0.69895
trained only on exemplars	0.61084	0.49049	0.54409	0.77010	0.65958	0.71057
trained on FN + exemplars guide features	0.65241	0.55960	0.60245			
frust† on combined	0.65702	0.59043	0.62195	0.73876	0.61397	0.67061
siblings‡ on FN	0.67244	0.54763	0.60365	0.64815	0.39088	0.48766
siblings, trained on combined	0.65991	0.60406	0.63075	0.76140	0.67713	0.71679
parents* on FN	0.67672	0.52790	0.59312	0.65250	0.38184	0.48176
parents, trained on combined	0.65920	0.60382	0.63029	0.76143	0.68317	0.72018
trained on semlink	0.44433	0.10533	0.17029			
trained on FN + semlink guide features	0.64671	0.54533	0.59171			
trained on FN+SemLink	0.655	0.3776	0.4791			
with SRL augmented spans and features	0.70550	0.53178	0.60644			

combined: FN + Exemplars training data

†: feature augmentation from the frustratingly easy DA paper

‡: for every feature f_i that fires for an argument a, fire an additional feature which is the conjunction: $(f_i \wedge \text{parent.frame} \wedge \text{parent.role} \wedge I_{hier})$ where parent.frame = parent(frame(a)). The parent's frame and role are obtained from the FN hierarchy. I_{hier} is an indicator to distinguish this feature from the regular conjunction features that use frame names and roles.

$\star$: fire the siblings feature‡ and an additional feature: $(f_i \wedge \text{frame} \wedge \text{frame.role} \wedge I_{hier})$

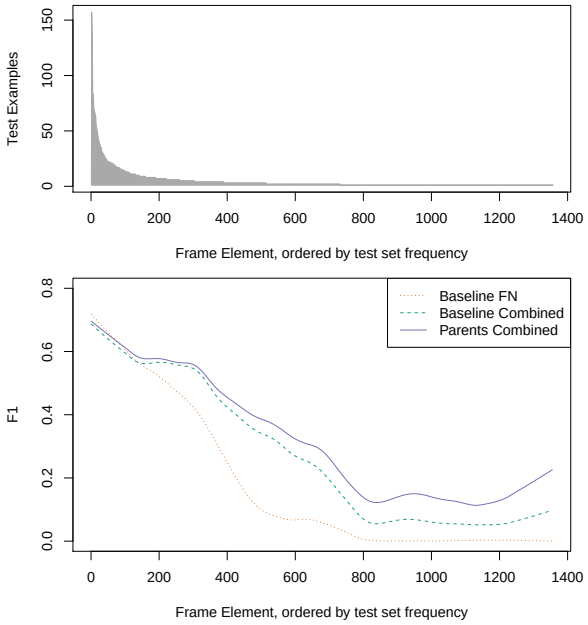

Figure 3: Count and F_1 for each frame element appearing in the test set. F_1 values have been smoothed with `loess`, with a smoothing parameter of 0.2.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proc. of COLING-ACL*, pages 86–90. Montreal, Quebec, Canada.

Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. 2014. Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56.

Paul Kingsbury and Martha Palmer. 2002. From TreeBank to PropBank. In *Proc. of LREC*, pages 1989–1993. Las Palmas, Canary Islands.

Dong C. Liu and Jorge Nocedal. 1989. On the Limited Memory BFGS Method for Large Scale Optimization. *Math. Program.*, 45(3):503–528.

Martha Palmer, Daniel Gildea, and Nianwen Xue. 2010. *Semantic Role Labeling*. Number 6 in Synthesis Lectures on Human Language Technologies. Morgan & Claypool, San Rafael, CA.

Matthew Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. *CoRR*, abs/1212.5701. URL `http://dblp.uni-trier.de/rec/bibtex/journals/corr/abs-1212-5701`.