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.[$_\text{S}^\text{NS}$ 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). [$_\text{S}^\text{NS}$ result]

1 Introduction

[$_\text{S}^\text{NS}$ 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, scenarios, and relationships with an inventory of **frames** ([$_\text{S}^\text{NS}$ 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
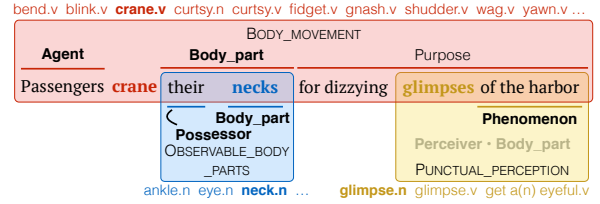
---

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

Figure 1: Example sentence from FrameNet full-text annotation. 3 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.

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 §2.2.

FrameNet 1.5 defines a structured hierarchy of over 1,000 frames associated with [$_\text{S}^\text{NS}$ #] English lexical predicates, and also provides annotations for [$_\text{S}^\text{NS}$ # 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—[$_\text{S}^\text{NS}$ #], comprising [$_\text{S}^\text{NS}$ #] words—are provided with **full-text** annotations, i.e. the sentence has been analyzed for all available frames. But a full [$_\text{S}^\text{NS}$ #]% 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

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.

it evokes. This is a form of semantic role labeling (SRL), a task introduced by Gildea and Jurafsky (2002) using a much earlier version of FrameNet. 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 $[^{\text{NS}}_S$ 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.$[^{\text{NS}}_S$ SOTA (without constraints)?]

2 Resources

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

Each frame in the Berkeley FrameNet lexicon is intended to represent a gestalt scene. The frame definition includes: a descriptive name; a set of **core roles** representing participants and props that are crucial to understanding the scene; a set of **non-core roles** such as circumstantial information (time, place, manner, purpose, etc.); an English textual description of the scene and how its roles relate to one another; and a set of English predicates that can evoke the scene. For example, the BODY_MOVEMENT frame has **Agent** and **Body_part** as its core roles; the frame description states, "This frame contains words for motions or actions an **Agent** performs using some part of his/her body." Lexical entries in this frame include verbs such as bend, blink, crane, and curtsy, plus the noun use of curtsy.

The frame 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). Among these kinds of frame-to-frame relations are:

- **Inheritance:** E.g., PUNCTUAL_PERCEPTION (e.g., glimpse.v) inherits from PERCEPTION_EXPERIENCE (e.g., see.v), which inherits from PERCEPTION. Other frames inheriting from PERCEPTION include SENSATION (e.g., sight.n) and BECOMING_AWARE (e.g., notice.v). Crucially, roles in inheriting (conceptually more specific) frames are mapped where they correspond to a role in the inherited frame: so PUNCTUAL_PERCEPTION.**Perceiver** links to PERCEPTION_EXPERIENCE.**Perceiver_passive**, which links to PERCEPTION.**Perceiver**, which links to SENSATION.**Perceiver_passive** and BECOMING_AWARE.**Cognizer**.

- **Subframe:** This indicates a subevent within a complex event. E.g., the CRIMINAL_PROCESS frame groups together subframes ARREST, ARRAIGNMENT, TRIAL, and SENTENCING. CRIMINAL_PROCESS.**Defendant**, for instance, is mapped to ARREST.**Suspect**, ARRAIGNMENT.**Defendant**, TRIAL.**Defendant**, and SENTENCING.**Convict**. Other salient participants in the complex event (such as the crime for which someone is arrested, tried, etc.) are similarly mapped via Subframe relations. This permits an inference that a person tried for a crime likely has been or will be arrested, arraigned, and sentenced for that crime.

In §3.5, 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 and how they relate syntactically to the predicate.

2.2 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. BODY_MOVEMENT, as described in the previous section, is evoked by *crane*, and 3 of its roles are filled by overt arguments: the 2 core roles (**Agent**, **Body_part**) happen to be filled by noun phrases (*passengers*, *their necks*), while the non-core role **Purpose** is filled by a prepositional phrase adjunct (*for dizzying glimpses of the harbor*). The frame defines 19 additional non-core roles, none of which have an argument in the example. In frame semantics, non-core roles are considered to be *conceptually* optional; core roles may or may not be *syntactically* optional, but if not locally specified they are expected to be available from context, or else implicit. For example, PUNCTUAL_PERCEPTION—evoked in this case by *glimpses*—is annotated as missing 2 of its core roles. A human listener would resolve the identity of the **Perceiver** from the wider context and the **Body_part** from world knowledge.

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.

[$^{NS}_S$ Genres, sizes]
[$^{NS}_S$ Annotation density: (proportion of tokens evoking a frame. breakdown by POS?)]

2.3 Exemplars

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

2.4 PropBank

PropBank (PB; Palmer et al., 2005) is a lexicon and corpus of predicate–argument structures that takes a shallower approach than FrameNet. Whereas FrameNet frames cluster lexical predicates that evoke similar kinds of scenarios (with the same kinds of roles), and these frames are organized in a network, PropBank frames are purely lexical and there are no formal relations between different predicates or their roles. PropBank does represent lexical ambiguity—e.g., the verb *order* is ambiguous in PropBank between order-v-01 "impelled action" and order-v-02 "request to be delivered"—but PropBank's sense distinctions are generally coarser-grained than FrameNet's.

Within sense-disambiguated PropBank frames, or **rolesets**, core roles are defined with textual descriptions and assigned numbers. E.g., order-v-02 defines: **A0** "orderer", **A1** "thing ordered", **A2** "benefactive, ordered-for", and **A3** "source". Following Dowty's (1991) theory of proto-roles, PropBank rolesets use **A0** for proto-agents and **A1** for proto-patients, but in general, there is much less consistency in interpretation of core roles across lexical predicates for PropBank than there is for FrameNet. Another difference is that PropBank's non-core roles—named **AM-***, such as **AM-PRP** for purposes—are not frame-specific.[4]

Despite all these differences, there is often a great deal in common between FrameNet-style and PropBank-style analyses, as should be apparent from comparing figure 1 and figure 2. The major benefit to PropBank is that it includes a large and comprehensively annotated corpus. We hypothesize that leveraging this large corpus indirectly can reap rewards for FrameNet-style SRL.

2.5 Illinois SRL system

2.6 SemLink

[$^{NS}_S$ limitations!]

3 Learning from multiple domains and representations

We use the model from SEMAFOR (Das et al., 2014), described in §3.1, as a starting point. We experiment with several domain adaptation techniques, augmenting the model's training data (§3.2) and feature space (§3.3–3.5).

3.1 Base model

The argument identification task is treated as a structured prediction problem. Let the classifica-

---

[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] Ellsworth et al. (2004) has a more extensive discussion of differences between PropBank's and FrameNet's conventions.

tion input be a dependency-parsed sentence $\mathbf{x}$, the token(s) p constituting the predicate in question, and the frame f evoked by p (as determined by frame identification). We use a heuristic procedure [$^{\text{S}}_{\text{T}}$ cite] for extracting candidate argument spans for the predicate; call this $spans(\mathbf{x}, p, f)$. $spans$ always includes a special span denoting an empty or non-overt role, denoted $\varnothing$. For each candidate span $a \in spans(\mathbf{x}, p, f)$, we extract a binary feature vector $\phi(a, \mathbf{x}, p, f)$. We describe the features in §**??**. We use a linear model, parametrized by the weight vector $\mathbf{w}$, to score a:

$$score_\mathbf{w}(a \mid \mathbf{x}, p, f, r) = \mathbf{w}^\top \phi(a, \mathbf{x}, p, f, r) \quad (1)$$

$score_\mathbf{w}$ models the compatibility of a candidate role–argument pair, and its parameters (feature weights) $\mathbf{w}$ are learned from data (§4.1).

At inference time, we use a **global classifier**. The global classifier chooses a joint assignment of all arguments of a frame, while respecting the following constraints:

1. a role may be assigned to at most one span, and
2. spans of overt arguments must not overlap.

Concretely, let a joint assigment be represented as a function $\mathbf{a} : roles(f) \rightarrow spans(\mathbf{x}, p, f)$, and let $\mathcal{A}$ be the set of all non-overlapping assignments. We give $\mathbf{a}$ the score

$$score_\mathbf{w}(\mathbf{a} \mid \mathbf{x}, p, f) = \sum_{r \in roles(f)} score_\mathbf{x}(\mathbf{a}(r) \mid \mathbf{x}, p, f, r)$$

$$(2)$$

And choose the joint assignment

$$args(\mathbf{x}, p, f) = \arg\max_{\mathbf{a} \in \mathcal{A}} score_\mathbf{w}(\mathbf{a} \mid \mathbf{x}, p, f, r) \quad (3)$$

Beam search, with a beam size of 100, is used to find this $\arg\max$.[5]

[$^{\text{M}}_{\text{K}}$ introduce this as a supervised domain adaptation problem? What is the source and what's the target?] [$^{\text{M}}_{\text{K}}$ Let D_{ft} represent the FT data and D_{ex} the exemplars?]

[$^{\text{NS}}_{\text{S}}$ Domain adaptation/multitask learning techniques]

---

[5]Recent work has improved upon global decoding techniques [$^{\text{S}}_{\text{T}}$ cite Täckström et al, TACL]. We expect such improvements to be complementary to the gains due to the added features and data reported here.

3.2 Augmenting the Training Data

3.3 Frustratingly Easy

Daumé III (2009) proposed a simple feature augmentation approach that was shown to work well in supervised domain adaptation scenarios, such as ours. Let $\mathcal{D}_{FT}$ be the full text data, and $\mathcal{D}_{ex}$ be the exemplar data. We introduce a domain feature $I_{domain}(\mathbf{x}) = \mathbf{1}\{\mathbf{x} \in \mathcal{D}_{FT}\}$, where $\mathbf{1}\{\cdot\}$ is the indicator function,

$$\mathbf{1}\{P\} = \begin{cases} 1, & \text{if } P \\ 0, & \text{otherwise.} \end{cases}$$

We expand the feature space by concatenating the original feature vector with a version of the feature vector that has been element-wise conjoined with the domain indicator:

$$\phi_{frust}(a, \mathbf{x}, p, f, r) = \begin{bmatrix} \phi(a, \mathbf{x}, p, f, r) \\ \phi(a, \mathbf{x}, p, f, r) \wedge I_{domain}(\mathbf{x}) \end{bmatrix} \quad (4)$$

The intuition here is that by replicating the features, we allow for each feature to contribute both "general" and domain-specific weights to the model depending on whether the feature behaves similarly in both domains or not. Since the "general" feature contributes to both domains, regularization will encourage the model to use the "general" version over the "specific" version of a feature whenever possible.

3.4 Guide Features

Another approach to introduce supervision for domain adaptation, which does not combine the labeled data from the two domains, to use a supervised model built on the source domain to augment features in the target domain. When training and testing in the target domain, the source domain model is effectively just a form of preprocessing to provide additional features, known as **guide features** (Johansson, 2013; Kong et al., 2014).[6] Formally: let M_s be the model built on the source domain (for instance, the PropBank data). For every target instance x, we introduce "guide" features which use the output $M_s(x)$ obtained by applying M_s on x, which consists of the role labels assigned to various text spans in x. Two types of guide features were used, one to indicate that a span s_x was

---

[6]This is related to the technique of model stacking, where successively richer models are trained by cross-validation on the same dataset Cohen and Carvalho (e.g., 2005); Nivre and McDonald (e.g., 2008); Martins et al. (e.g., 2008).

assigned a role and the other encoding the role label r_g itself. In the case where M_s produces labels that belong to the same schema as the target domain (for instance, the exemplars use the same schema as the FT annotations), we use an additional feature $f_{match}(r_t, r_g)$ to indicate that the 'guide' role label r_g of the span s_x is the same as it's true label r_t.

3.5 Type-level hierarchy features

[^M_K NSS: notation for the frame/role/features etc?]
[^M_K How many total types of relations? Cleanup writeup based on NSS's notation.]

Frames in FrameNet are connected to each other by relations such as inheritance, temporal ordering, causality. For instance, the frame ROBBERY inherits from the more abstract frame COMMITTING_CRIME, the frame FALL_ASLEEP is preceeded by the frame BEING_AWAKE. The roles of related frames have also been mapped to indicate the correspondence between them: ROBBERY.**Perpetrator** is mapped to COMMITTING_CRIME.**Perpetrator**, which in turn maps to MISDEED.**Wrongdoer**. Frames and roles that are far apart in this hierarchy are less related than say neighbours. This hierarchy can be exploited to share information across related roles, thereby benefiting the roles that have few annotations [^M_K say something about a greater variety of contexts is available for each role]. A simple mechanism to share information is via shared model parameters between related roles. Towards this, we define two types of hierarchical features: (1) 'siblings', where roles that have a common parent share features. For every feature f_i that fires for an argument a, we add a feature which is the conjunction: $(f_i \wedge \text{parent.frame} \wedge \text{parent.role} \wedge I_{hier})$ where parent.frame = parent(frame(a)). I_{hier} is an indicator to distinguish this feature from the regular conjunction features in Semafor that use frame names and roles. (2) 'parents+siblings', where roles share features with their parents and siblings. In addition to the 'siblings' feature, we fire the following conjunction: $(f_i \wedge \text{frame} \wedge \text{frame.role} \wedge I_{hier})$. We use only the Inheritance and Subframe relations between the roles (§2.1), of which there are 4138 and 589 respectively.

We found that using more than one level of the hierarchy, does not produce any improvements in the performance at the cost of increased computation due to the greater number of features.

[^M_K Write about the number of features in all models somewhere.. Baseline: 2709128, Combined training: 12972404, Frust-easy: 15681531, 'Parents+siblings': 34282958]

4 Experiments

All of our experiments use the same form of regularization, condition on the same oracle frame predictions, and syntactic preprocessing[$^{NS}_S$ does this match Dipanjan's latest experiments?], and use beam search with a beam size of 100 for joint decoding of the test data. Automatic syntactic dependency parses from MSTParserStacked (Martins et al., 2008) are used, as in Das et al. (2014).

4.1 Learning

[^S_T TODO: local classifier is only used in my SVM version, not in Das's log-linear version.] We define the **local classifier** as:

$$arg(\mathbf{x}, p, f, r) = \underset{a \in spans(\mathbf{x}, p, f)}{\arg\max}\ score_{\mathbf{w}}(a \mid \mathbf{x}, p, f, r)$$

(5)

We use the local classifier during training.

[^S_T talk about learning here?] [^S_T something about L2 regularization]

We have made several modifications to SE-MAFOR's training in order to 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 these changes for all systems, including the baseline. While the impact on full-text performance is negligible, these changes enabled us to run more experiments with the larger exemplar dataset and expanded feature space.[7]

The details of squared hinge loss are as follows. Let $(x^{(i)}, y^{(i)})$ be the ith training example. Then the

---

[7]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[$^{NS}_S$ say something about hardware this was tested on?].

structured hinge loss on the ith example is given by:

$$Hinge_{\mathbf{w}}(i) =$$
$$\max_{y} \left\{ \mathbf{w}^{\top} \mathbf{f}(x^{(i)}, y) + \text{cost}(y, y^{(i)}) \right\} - \mathbf{w}^{\top} \mathbf{f}(x^{(i)}, y^{(i)})$$
$$\tag{6}$$

and squared hinge loss is:

$$SquaredHinge_{\mathbf{w}}(i) = Hinge_{\mathbf{w}}(i)^2. \tag{7}$$

In our experiments, we use $\text{cost}(y, y^{(i)}) = \mathbf{1}\{y \neq y^{(i)}\}$ [$_{\text{T}}^{\text{S}}$ define][8].

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

[$_{\text{S}}^{\text{NS}}$ 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)]

[$_{\text{S}}^{\text{NS}}$ tuning regularizer for all experiments] We tune the ℓ_2 regularization parameter λ on the held-out set. We searched over the following values: 10^{-5}, 10^{-7}, 10^{-9}, 10^{-12} (note that our loss is normalized). We also use the performance on the held-out set to determine the stopping criterion for the stochastic optimization.

4.2 Evaluation

The argument identification performance of the various methods is compared on two different test sets. (1) Full-text: the FrameNet 1.5 FT test split that was used in the evaluation in). This data consists of sentences from 23 documents. (2) Exemplars: a randomly sampled set of sentences from the exemplars data, with approximately the same number of targets as the FT test set. Statistics of both the test sets are given in Table 1. While the FT test set represents the benchmark set for evaluating the performance of a frame semantic role labeling system, the exemplars test set being from a different distribution of text, give us an indication of how well a model generalizes. We present precision, recall, and F1-measure microaveraged across the test instances.

[$_{\text{K}}^{\text{M}}$ Analyze the sparsity of the models? Which features have highest importance?]

---

[8]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[$_{\text{S}}^{\text{NS}}$ or: failed to improve?].

	Full-Text	Exemplars
Sentences in training	2,780	145,838
Roles in train data	15,019	137,511
Roles (types) in training	2,644	4,821
Sentences in test data	2,420	4,132
Roles in test data	4,458	4,132
Roles (types) in test data	1,420	1,224
Unseen frames (types) in test data	46	0
Roles (types) in unseen frames	178	0
Unseen roles (types) in test data	289	38
Unseen roles (types) in test w.r.t (FT+Exemplars)	103	32

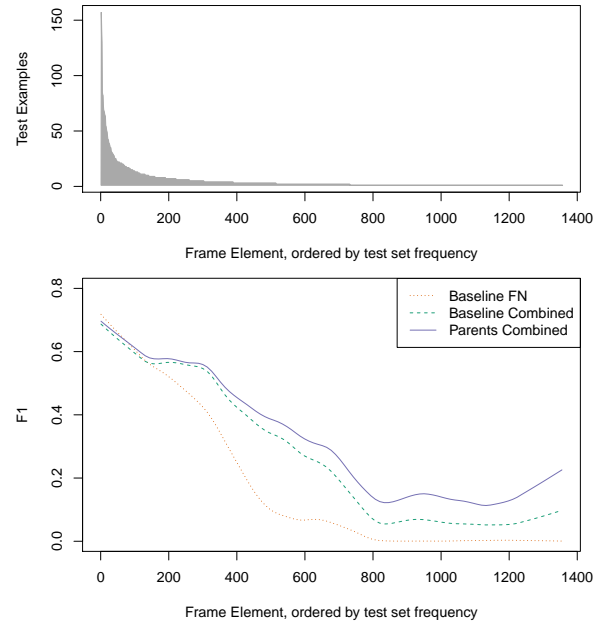**Table 1:** Characteristics of the training and test data

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.

4.3 Results

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

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

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

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

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

5 Related Work

[$_{\text{S}}^{\text{NS}}$ Dipanjan's other papers; mention other PB SRL work?; anything using SemLink or combining resources for SRL? see (Bonial et al., 2013, §4)]

[$_{\text{S}}^{\text{NS}}$ be sure to cite: (Shi and Mihalcea, 2005)

Additional Resource	Training Configuration (Features)	Full-Text			Exemplars		
		P	R	F_1	P	R	F_1
(Baseline)	FT (Basic)	65.57	53.82	59.12	62.63	37.65	47.03
FN Hierarchy	FT (siblings)	67.24	54.76	60.36	64.81	39.09	48.77
	FT (siblings+parents)	67.67	52.79	59.31	65.25	38.18	48.18
Exemplars	Exemplars $\xrightarrow{\text{guide}}$ FT	65.24	55.96	60.24	67.71	48.08	56.23
	FT+Exemplars (Basic)	66.06	58.23	61.9	75.44	65.11	69.89
	FT+Exemplars (EasyAdapt)	65.70	59.04	62.19	73.88	61.40	67.06
SemLink	SemLink $\xrightarrow{\text{guide}}$ FT	64.67	54.53	59.17	60.95	38.92	47.50
	FT+SemLink	65.50	37.80	47.90	57.15	20.80	30.50
PB-SRL	FT (PB-SRL)	64.96	54.83	59.47	61.38	39.14	47.80

Table 2: Results on two test sets: Baseline vs. individual other resources. Precision, recall, and F_1 are given as percentages.

Training Configuration (Features)	Full-Text			Exemplars		
	P	R	F_1	P	R	F_1
FT+Exemplars (Hier: siblings)	66.00	60.40	63.07	76.14	67.71	71.70
FT+Exemplars (PB-SRL)	67.36	58.79	62.80	77.15	65.47	70.83

Table 3: Combining best techniques across resources [$_S^{NS}$ TODO]

(multiple resources), (Matsubayashi et al., 2009) (hierarchy)]

[$_S^{NS}$ multitask learning?]

6 Conclusion

[$_S^{NS}$ overall findings]

[$_S^{NS}$ future work: testing ground for improvements to PB (Bonial et al., 2014) and SemLink (Bonial et al., 2013); automatic mappings between resources]

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.

Claire Bonial, Julia Bonn, Kathryn Conger, Jena D. Hwang, and Martha Palmer. 2014. PropBank: semantics of new predicate types. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proc. of LREC*, pages 3013–3019. Reykjavík, Iceland.

Claire Bonial, Kevin Stowe, and Martha Palmer. 2013. Renewing and revising SemLink. In *Proc. of the 2nd Workshop on Linked Data in Linguistics (LDL-2013): Representing and linking lexicons, terminologies and other language data*, pages 9–17. Pisa, Italy.

William W. Cohen and Vitor R. Carvalho. 2005. Stacked sequential learning. In *Proc. of IJCAI*, pages 671–676. Edinburgh, Scotland, UK.

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

Hal Daumé III. 2009. Frustratingly easy domain adaptation. *arXiv preprint arXiv:0907.1815*.

David Dowty. 1991. Thematic proto-roles and argument selection. *Language*, 67(3):547–619.

Michael Ellsworth, Katrin Erk, Paul Kingsbury, and Sebastian Padó. 2004. PropBank, SALSA, and FrameNet: How design determines product. In *Proc. of the LREC 2004 Workshop on Building Lexical Resources from Semantically Annotated Corpora*, pages 17–23. Lisbon, Portugal.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Richard Johansson. 2013. Training parsers on incompatible treebanks. In *Proc. of NAACL-HLT*, pages 127–137. Atlanta, Georgia, USA.

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

Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer, and Noah A. Smith. 2014. A dependency parser for tweets. In *Proc. of EMNLP*, pages 1001–1012. Doha, Qatar.

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

André F. T. Martins, Dipanjan Das, Noah A. Smith, and Eric P. Xing. 2008. Stacking dependency parsers. In *Proc. of EMNLP*, pages 157–166. Honolulu, Hawaii.

Yuichiroh Matsubayashi, Naoaki Okazaki, and Jun'ichi

Tsujii. 2009. A comparative study on generalization of semantic roles in FrameNet. In *Proc. of ACL-IJCNLP*, pages 19–27. Suntec, Singapore.

Joakim Nivre and Ryan McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proc. of ACL-HLT*, pages 950–958. Columbus, Ohio, USA.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The Proposition Bank: an annotated corpus of semantic roles. *Computational Linguistics*, 31(1):71–106.

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.

Josef Ruppenhofer, Michael Ellsworth, Miriam R. L. Petruck, Christopher R. Johnson, and Jan Scheffczyk. 2010. FrameNet II: extended theory and practice. URL `https://framenet2.icsi.berkeley.edu/docs/r1.5/book.pdf`.

Lei Shi and Rada Mihalcea. 2005. Putting pieces together: combining FrameNet, VerbNet and WordNet for robust semantic parsing. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing: Proceedings of the 6th International Conference (CICLing 2005)*, volume 3406 of *Lecture Notes in Computer Science*, pages 100–111. Springer-Verlag, Berlin.

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`.