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 accuracy and domain robustness of models trained on them. However, low-resource tasks may benefit from exploiting *out-of-domain* 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. 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). Experiments demonstrate that some of these resources are valuable for more accurate and more robust frame SRL, with the best model achieving a 3.95 increase in F_1 over the state of the art. [$^{NS}_{S}$ main quantitative result].

1 Introduction

Sparsity of data resources is a challenge for many computational semantics tasks [$^{S}_{T}$ cite?]. 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** (such as SHOPPING, SCARCITY, and SHOOT_PROJECTILES). 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 §2.2.

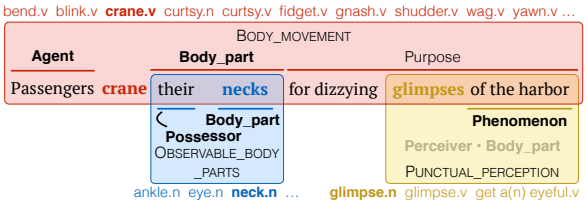


Figure 1: Example sentence from FrameNet full-text annotation. 3 frames and their arguments are shown: BODY_MOVEMENT is evoked by *crane*, OBSERVABLE_BODY_PARTS by *necks*, and PUNCTUAL_PERCEPTION by *glimpse*. (Further, *harbor* is annotated as evoking the LOCALE_BY_USE frame and doubles as its sole argument.) Horizontal lines representing argument spans are labeled with role names.

FrameNet 1.5 defines a structured taxonomy of over 1,000 frames associated with 12,000 English lexical predicates, and also provides annotations for over 175,000 attestations of these frames/predicates in corpora, annotated in context with their arguments. A rather small number of sentences (about 5,000) are provided with **full-text** annotations, i.e. each sentence has been analyzed for all available frames. But the bulk of sentences in FrameNet—the lexicographic **exemplars**—are annotated for only one frame per sentence, and have thus far not been exploited successfully for the SRL phase of the contemporary frame-semantic parsing task. Here, we seek to leverage these exemplar sentences as well as the (type-level) hierarchical structure of the FrameNet lexicon.

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

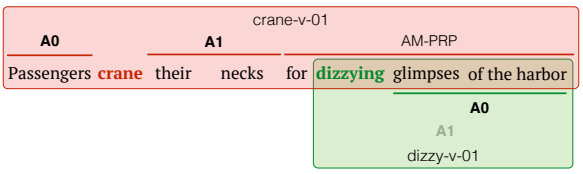


Figure 2: Ideal PropBank annotations for verbs. Though PB uses lexical frames rather than deep frames, there are clear similarities to the FrameNet annotations in figure 1.

In this paper, we address the **argument identification** subtask of finding and labeling arguments given a predicate in context and the frame 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 [$^{NS}_{S}$ 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.[$^{NS}_{S}$ SOTA (without constraints)?]

2 Resources

[$^{NS}_{S}$ incl. data analysis of differences from full-text]
[$^{NS}_{S}$ w/in each mention: genre/overall vocabulary; coverage and distributions of predicates, FE labels; oracle coverage of FT test]

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

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

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

Beginning with the SemEval-2007 shared task on FrameNet analysis (**?**), frame-semantic parsers have been 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.

The full-text documents represent a mix of genres, prominently including travel guides and bureaucratic reports about weapons stockpiles. Statistics for the full-text corpus appear in table 1.

[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.

[$^{NS}_{S}$ Annotation density: (proportion of tokens evoking a frame. breakdown by POS?)]

2.3 Exemplars

Conceived primarily as a lexicography project, most FrameNet annotations serve to illustrate the argument structure potential of particular predicates. When predicates are added to a frame, a large set of source corpora (primarily, the British National Corpus) covering a wide range of genres is searched for various syntactic patterns, and a lexicographer identifies a selection, or **subcorpus**, of sentences illustrating the predicate's behavior (**?**). The subcorpus sentences are then annotated, but *only with respect to the predicate in question*. These singly-annotated sentences are called lexicographic **exemplars**.

The subset of exemplars containing argument annotations is described in table 1. Relative to the full-text dataset, the exemplar dataset contains an order of magnitude more frame annotations and two orders of magnitude more sentences annotated. Because we are conditioning on the identified frame, the fact that the exemplar dataset has just one annotated frame per sentence is not a concern in SRL. However, the rate of overt arguments per frame is noticeably higher for exemplars, which potentially biases the model's tendency to predict certain kinds of arguments in a way that is not statistically representative of a natural corpus.

The exemplar sentences formed the basis of early studies of frame-semantic role labeling (e.g., Gildea and Jurafsky, 2002; **?**; **?**; **?**). We deem it worthwhile to (a) investigate whether the exemplars can be used to improve performance on the full-text evaluation, compensating for the scarcity of full-text training data, and (b) to evaluate SRL performance on a held-out set of exemplars, given that these exemplars represent a much broader range of genres, frames, and predicates than the full-text data. This second evaluation, we believe, will give a useful indication of the robustness of the SRL model.

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.

Very little data is annotated with both PropBank and FrameNet analyses. Therefore, to bridge between the PropBank and FrameNet corpora, we explore two approaches: (a) running a PropBank-trained semantic role labeler on the FrameNet data as an additional form of preprocessing; and (b) leveraging SemLink (Bonial et al., 2013), a partial and semi-automatic augmentation of the PropBank corpus's roleset annotations with mappings to FrameNet and VerbNet.

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).

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

3.1 Base model

The argument identification task is treated as a structured prediction problem. Let the classification 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 [S 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 §**??**.[S T are we going to describe the baseline semafor features?][NS S i suggest just mention the kinds of things features look at—POS, syntax, etc.—and that many of them are at multiple levels of granularity (role-specific, role name–specific)] We use a linear model, parametrized by the weight vector $\mathbf{w}$, to score a:

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

$score_{\mathbf{w}}$ 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.

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

$$score_{\mathbf{w}}(\mathbf{a} \mid \mathbf{x}, p, f) = \sum_{r \in roles(f)} score_{\mathbf{x}}(\mathbf{a}(r) \mid \mathbf{x}, p, f, r) \qquad (2)$$

and choose the joint assignment:

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

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

[5] Recent work has improved upon global decoding techniques (**?**). We expect such improvements to be complementary to the gains due to the added features and data reported here.

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

[NS S Domain adaptation/multitask learning techniques]

3.2 Augmenting the Training Data

3.3 Frustratingly Easy

? 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 indicator $I_{\{\mathbf{x}\in\mathcal{D}_{FT}\}}$, where $I_{\{P\}}$ is the indicator function, with value 1 if P is true, 0 otherwise. 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:

$$\phi_{frust}(a,\mathbf{x},p,f,r) = \begin{bmatrix} \phi(a,\mathbf{x},p,f,r) \\ \phi(a,\mathbf{x},p,f,r)\ \&\ I_{\{\mathbf{x}\in\mathcal{D}_{FT}\}} \end{bmatrix}$$

[S T this "vector concatenation" view is sort of inconsistent with the "adding features" view in the guide/hierarchy sections]

The intuition 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 domain-specific version of a feature whenever possible.

3.4 Guide Features

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

In our case, we treat the full text annotations as our target domain, and experiment with using PropBank, SemLink, and the exemplars data as source domains. Each of the three source models produces SRL-style output, where predicates are assigned frames or rolesets, and for each predicate, spans are assigned role labels. But they differ in the labels used for roles. [S T talk about Illinois SRL (**?**)[7], whatever-we-did-with-SemLink here?]

Formally, let M_s be the model built on the source domain (for instance, the PropBank data). For every target domain sentence $\mathbf{x}$, we introduce "guide" features which use the output $M_s(\mathbf{x})$ obtained by applying M_s on $\mathbf{x}$, which consists of the role labels assigned to various text spans in $\mathbf{x}$. Two types of guide features were used: one indicates that a span a was assigned *any* role, and the other encodes 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 [S T our set of source domains is small enough that we can enumerate which ones this applies to. Is it exemplars + SemLink?]), we use an additional feature $\phi_{match}(r_t, r_g)$ to indicate that the 'guide' role label r_g of the span a is the same as it's true[S T you mean "target"?] 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, and 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]. We say that the *parent* of a role is one that has either

[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).

[7] http://cogcomp.cs.illinois.edu/page/software_view/SRL

the **Inheritance** or **Subframe** relation to it (§2.1). There are 4138 **Inheritance** and 589 **Subframe** links between role types in FrameNet 1.5.

A simple mechanism to share information is via shared model parameters between related roles. Towards this, we experiment with two variations of hierarchical feature types:

- **siblings**: Roles that have a common parent share features. For every feature $\phi_i(a,\mathbf{x},p,f,r)$, we add a new feature which is the conjunction: "sib" $\wedge\, \phi_i(a,\mathbf{x},p,f,r) \wedge parent(r)$.
- **parent+siblings**: Roles share features with their parent and siblings. For every feature $\phi_i(a,\mathbf{x},p,f,r)$, we add a two new features: "par+sib" $\wedge\, \phi_i(a,\mathbf{x},p,f,r) \wedge parent(r)$, and "par+sib" $\wedge\, \phi_i(a,\mathbf{x},p,f,r) \wedge r$.

We experimented with using more than one level of the hierarchy (grandparents, e.g.), but found that it does not produce any improvements in the performance, yet increased computation cost due to the greater number of features.

4 Experiments

All of our experiments use the same form of regularization, condition on the same oracle frame predictions, and syntactic preprocessing[NS 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

Following SEMAFOR, we train using a **local** objective instead of using a global classifier. In other words, we treat each role and span pair as an independent training instance. But we have made several modifications to SEMAFOR's training in order to speed up experiments:

- We minimize squared structured hinge loss (defined below) instead of a log-linear loss. Using hinge loss, there is no longer a need to calculate a partition function. Gradients, and hence parameters, are sparser than in a log-linear model, as they only depend on the correct span and the predicted span.
- We use the online optimization method AdaDelta (Zeiler, 2012) with minibatches, instead of the batch method L-BFGS (Liu and Nocedal, 1989). We use minibatches of size 4,000 on the full text data, and 40,000 on the exemplar data.

We 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.[8]

The details of squared hinge loss are as follows. Let $((\mathbf{x},p,f,r),a)$ be the i^{th} training example. Then the structured hinge loss on the i^{th} example is given by:

$$Hinge_{\mathbf{w}}(i) = \max_{a'} \{\mathbf{w}^\top \phi(a',\mathbf{x},p,f,r) + \text{cost}(a',a)\} - \mathbf{w}^\top \phi(a,\mathbf{x},p,f,r)$$

and squared hinge loss is:

$$SqHinge_{\mathbf{w}}(i) = Hinge_{\mathbf{w}}(i)^2. \quad (4)$$

We use $\text{cost}(a',a) = I_{\{a' \neq a\}}$.[9]

We learn $\mathbf{w}$ by minimizing the ℓ_2-regularized average loss on the dataset:

$$\mathbf{w}^* = \arg\min_{\mathbf{w}} \frac{1}{N} \sum_{i=1}^{N} \{SqHinge_{\mathbf{w}}(i)\} + \frac{1}{2}\lambda \|\mathbf{w}\|_2^2 \quad (5)$$

4.2 Preprocessing the Data

Full-Text. For the FT annotations, we use the same FrameNet 1.5 data and train/test splits as Das et al. (2014), without any of our own processing.

Exemplars. We processed the exemplars data (§2.3) to remove sentences which had no role annotations (under the assumption that these are likely to be incomplete annotations). Out of a total of 145838 sentences, ≈4000 had only frame annotations. We further removed duplicate sentences that already appear in the FT data. In the argument annotations, we merged spans which were adjacent and had the same argument label.

[8] 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 say something about hardware this was tested on?].

[9] 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 or: failed to improve?].

SemLink. SemLink (see §2.4) contains two types of mappings. The *sense-level mappings* give correspondences between the concepts from each resource—i.e., between 'frames' from FN, 'role-sets' from PB, and 'roles' from VN. Since they map different interpretations and granularities of concepts, the sense-level mappings may be one-to-one, one-to-many, or many-to-many. PropBank and FrameNet are mapped indirectly via VerbNet. Second, SemLink provides some *token-level parallel annotations* for the 3 representations in a subset of the PB-WSJ text: hereafter SL-WSJ.

We focus on using token-level SemLink version 1.2.2c annotations as a (disambiguated) mapping from PB to FN tokens. Of the available 74,977 SL-WSJ verbs, a majority cannot be mapped to FN frames for various reasons. Around 31% of the predicates have the frame label `IN` ("indefinite") where the mapping from VerbNet to FrameNet is ambiguous. About 20% of the instances are labeled `NF` ("no frame"), indicating a coverage gap in FrameNet. 21% of verbs have frame labels but no frame element annotations. Most of these are predicates with modifier arguments. Other arguments pointed to null anaphora that could not be resolved to overt arguments. This leaves 15,323 mappable instances with at least one overt argument, or 20% of SL-WSJ verbs. This is a very small subset of the entire PB annotated data.

[$_{\text{S}}^{\text{NS}}$ OntoNotes PropBank preprocessing (NLTK)?]

4.3 Hyperparameter Tuning

We tune the ℓ_2 regularization parameter λ on the FT dev set, searching 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 FT dev set to determine the stopping criterion for the stochastic optimization. The FT dev set was used only to tune parameters and not as part of the model construction.

4.4 Evaluation

A complete semantic parsing system involves two main steps: frame identification and argument identification. Since the focus of this work is only argument identification, we assume that the gold-standard frames are given and evaluate the models on their role labeling performance. The baseline model from Das et al. (2014) that we compare against, is the current state-of-the-art for argument identification (i.e frame SRL).

	Full-Text		**Exemplars**	
	train	*test*	*train*	*test*
Sentences	2,780	2,420	137,515	4,132
Frames	15,019	4,458	137,515	4,132
Overt args	25,918	7,210	278,985	8,417
TYPES				
Roles	2,644	1,420	4,821	1,224
Unseen frames *vs. train:*		46		0
Roles in unseen frames *vs. train:*		178		0
Unseen roles *vs. train:*		289		38
Unseen roles *vs. combined train:*		103		32

Table 1: Characteristics of the training and test data. (These statistics exclude the development set, which contains 4,463 frames over 746 sentences.)

The 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 Das et al. (2014). 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 the lower half of Table 1. The FT test set has 289 unseen role types, which is much higher than the 38 in the exemplars test set. There are no unseen frame types in the exemplars test data, whereas the FT test has 46 of them. These differences are due to the manner in which the train and test splits were created, with document-level splits being used for FT and sentence-level splits for the exemplars. The last row of Table 1 shows the unseen role types faced by a model that was built on both the FT and exemplars training data. In the FT test set it is 103, lower by ≈190 than what a FT-only model will see, thus leading us to expect that a model that combines both sets of data will certainly benefit in performance.

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, gives us an indication of how well a model generalizes.

4.5 Results

[$_{\text{S}}^{\text{NS}}$ comparison to prior work (baseline, best result). args+frames score vs. args only] [$_{\text{S}}^{\text{NS}}$ be sure to mention: gold frames] We present precision, recall, and F_1-measure microaveraged across the test instances in Table 2, for all the approaches that we tried. The first column classifies the approaches based on what resource we use and the second column indicates what training data was used. For each resource (one multi-row in the table) we show

results obtained by various methods of combining it with the FT data. The marker '$\xrightarrow{\text{guide}}$' on some of the methods refers to the feature augmentation discussed in §3.4, 'Hier' represents the hierarchical features from §3.5 and 'EasyAdapt' refers to the frustratingly easy domain adaptation model described in §3.3.

The baseline F_1 on argument identification published in Das et al. (2014) on the FT test set includes adding 'pseudo' scores for getting gold frames right. However, this does not give us a true picture of the performance, hence we choose not to add frame-level scores while computing F_1. In Table 2, the first row corresponds to the state-of-art F_1 from Das et al. (2014).

The first resource that we consider is 'FN Hierarchy'. We find that adding the sibling-level hierarchical features to the baseline FT model improves F_1 by 1.2 and 1.7 points on FT-test and exemplars-test respectively, with benefits to both precision and recall. The 'siblings+parents' features which consider two levels of the hierarchy produce unnoticeable benefits, suggesting that higher levels in the hierarchy can be too general [$^{\text{M}}_{\text{K}}$ give example?] and cause very dissimilar roles firing the same hierarchy features.[10]

SemLink, as we remarked earlier is a noisy resource and we see a confirmation for this observation in the performance of the (FT+SemLink) model, which drops F_1 by a whopping 11.2 and 16.5 points below baseline on FT-test and exemplars, respectively. The guide features however modulate the influence of the SemLink annotations, giving a minor increase over the baseline.

With the exemplars resource, we find that the guide features give us a modest improvement of 1.12 in F_1 on FT-test, while using it as gold-standard training data results in a bigger increase of 2.8 points. On the exemplars-test we observe a similar trend, where the (FT+Exemplars) model delivers a massive increase of ≈23 F_1-points and a much smaller increase is seen with the guide features. This contrasts with what we observed for SemLink. The frustratingly easy domain adaptation approach to incorporate the Exemplars further improves the F_1 by a minor 0.3 on the FT-test and a decent 2.8 on the exemplars-test.

[10] We also tried adding grandparents and found only minor improvements. Using an expert-pruned hierarchy with relations that are most likely to help can give greater benefits, and is beyond the focus of this work.

Finally, using the PB-SRL data in the form of guide features also results in small improvements over the baseline. Overall we observe that an additional resource that is very similar to the original resource helps more as training data than as a 'guide model'. Whereas the guide features help more when the additional resource is either unreliable (like SemLink) or too distinct (like PB). To summarize the results, we find that adding the sibling-level hierarchical features, using Exemplars as gold-standard training data and incorporating PB-SRL data in the form of guide features all help in improving the performance over the baseline on both FT-test and exemplars-test.

4.6 Combining all resources

We pick the techniques that performed best across the resources and combine them in two models. Table 3 shows the results on these. Both models use the optimal training data configuration: (FT+Exemplars). The first one integrates the FN Hierarchy using the 'siblings' features and the second integrates PB-SRL data in the form of guide features. These two models outperform the models discussed in Table 2. On both test sets, we find that the model using the hierarchy achieves the best recall, and the PB-SRL guide features result in the best precision. Our best result of 63.07 on the FT test set beats the baseline (59.12) by 3.95 F points.

4.7 FE-level evaluation

The results so far have discussed the overall improvement in F_1-score, but do not present a detailed picture of how we gain from the additional coverage. Towards this, we present a frame-element i.e role type level analysis comparing the best results with the baseline. The first plot in Figure 3a shows the frequency of all roles that appear in the FT-test set. Figure 3b shows the F_1 per role-type, for the baseline and the two models from Table 3. In both plots, each point on the x-axis represents one role, with the roles sorted in the decreasing order of their frequency. The F_1 curves clearly show that our models achieve the best improvements for the rarer roles and are at par with the baseline on the frequent roles.

[$^{\text{M}}_{\text{K}}$ Analyze the sparsity of the models? Which features have highest importance?] In addition to the performance, we also show the sizes of the various models in the column 'Millions of features'. All the models are quite sparse however, with ≈60-70% of the features assigned a weight of zero. The

Additional Resource	Training Configuration (Features)	Millions of Features	Full-Text			Exemplars		
			P	R	F_1	P	R	F_1
(Baseline)	FT (Basic)	2.7	65.57	53.82	59.12	62.63	37.65	47.03
FN Hierarchy	FT (siblings)	5.4	67.24	54.76	60.36	64.81	39.09	48.77
	FT (siblings+parents)	8.5	67.67	52.79	59.31	65.25	38.18	48.18
SemLink	SemLink $\xrightarrow{\text{guide}}$ FT	3.0	64.67	54.53	59.17	60.95	38.92	47.50
	FT+SemLink	5.0	65.50	37.80	47.90	57.15	20.80	30.50
Exemplars	Exemplars $\xrightarrow{\text{guide}}$ FT	3.5	65.24	55.96	60.24	67.71	48.08	56.23
	FT+Exemplars (Basic)	13[$^{\text{NS}}_{\text{S}}$.?]	66.06	58.23	61.90	75.44	65.11	69.89
	FT+Exemplars (EasyAdapt)	16[$^{\text{NS}}_{\text{S}}$.?]	65.70	59.04	62.19	73.88	61.40	67.06
PB-SRL	PB-SRL $\xrightarrow{\text{guide}}$ FT	3.6	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)	Millions of Features	Full-Text			Exemplars		
		P	R	F_1	P	R	F_1
FT+Exemplars (Hier: siblings)	34	66.00	60.40	63.07	76.14	67.71	71.70
PB-SRL $\xrightarrow{\text{guide}}$ FT+Exemplars	17	67.36	58.79	62.80	77.15	65.47	70.83

Table 3: Combining best techniques across resources [$^{\text{NS}}_{\text{S}}$ TODO]

larger models take roughly 6 times as long as the baseline to train.

5 Related Work

[$^{\text{NS}}_{\text{S}}$ Dipanjan's other papers (focusing on different parts of the task)]

[$^{\text{NS}}_{\text{S}}$ be sure to cite: (Shi and Mihalcea, 2005) (multiple resources), (Matsubayashi et al., 2009) (hierarchy vs. role names), (**?**) (multiple resources + levels of generalization), (**?**) (semi-supervised)]

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

6 Conclusion

We have empirically shown that auxiliary semantic resources can benefit the challenging task of frame-semantic role labeling. Complementary gains on the standard full-text evaluation are achieved by including the FrameNet exemplars in the training data and by grouping features according to the FrameNet hierarchy. There are smaller, but nevertheless promising, signs that data annotated with the PropBank scheme can be leveraged as well. In addition to making SRL more *accurate*, a parallel evaluation on a sample of exemplars suggests the improved models are also more *robust*, capturing a broader swath of the English language.

We are optimistic that future improvements to lexical semantic resources, including planned changes for PropBank (Bonial et al., 2014) and SemLink (Bonial et al., 2013), will lead to further gains in this task. Moreover, the techniques discussed here could be further explored using semi-automatic mappings between lexical resources (such as UBY; **?**), and correspondingly, this task could be used to extrinsically validate those mappings.

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.

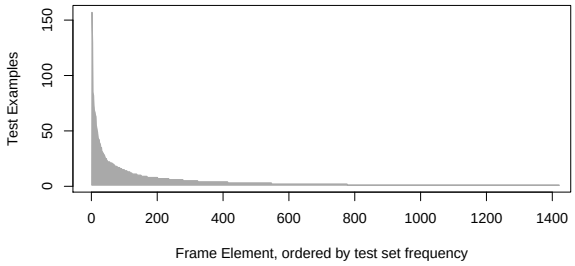
Claire Bonial, Julia Bonn, Kathryn Conger, Jena D. Hwang, and Martha Palmer. 2014. PropBank: semantics of new predicate types. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, *Proc. of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

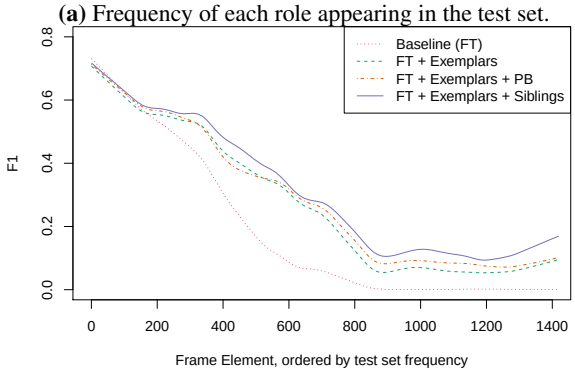
William W. Cohen and Vitor R. Carvalho. 2005. Stacked sequential learning. In *Proc. of IJCAI*, pages 671–676. Edinburgh, Scotland, UK.

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*.



(a) Frequency of each role appearing in the test set.



(b) F_1 of the best methods compared with the baseline.

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

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`.