Semi-Supervised Frame-Semantic Parsing for Unknown Predicates

Dipanjan Das and **Noah A. Smith**
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{dipanjan,nasmith}@cs.cmu.edu

Abstract

We describe a new approach to disambiguating semantic frames evoked by lexical predicates previously unseen in a lexicon or annotated data. Our approach makes use of large amounts of unlabeled data in a graph-based semi-supervised learning framework. We construct a large graph where vertices correspond to potential predicates and use label propagation to learn possible semantic frames for new ones. The label-propagated graph is used within a frame-semantic parser and, for unknown predicates, results in over 15% absolute improvement in frame identification accuracy and over 13% absolute improvement in full frame-semantic parsing F_1 score on a blind test set, over a state-of-the-art supervised baseline.

1 Introduction

Frame-semantic parsing aims to extract a shallow semantic structure from text, as shown in Figure 1. The FrameNet lexicon (Fillmore et al., 2003) is a rich linguistic resource containing expert knowledge about lexical and predicate-argument semantics. The lexicon suggests an analysis based on the theory of frame semantics (Fillmore, 1982). Recent approaches to frame-semantic parsing have broadly focused on the use of two statistical classifiers corresponding to the aforementioned subtasks: the first one to identify the most suitable semantic frame for a marked lexical predicate (*target*, henceforth) in a sentence, and the second for performing semantic role labeling (SRL) given the frame.

The FrameNet lexicon, its exemplar sentences containing instantiations of semantic frames, and full-text annotations provide supervision for learning frame-semantic parsers. Yet these annotations lack coverage, including only 9,300 annotated target types. Recent papers have tried to address the coverage problem. Johansson and Nugues (2007) used WordNet (Fellbaum, 1998) to expand the list of targets that can evoke frames and trained classifiers to identify the best-suited frame for the newly created targets. In past work, we described an approach where latent variables were used in a probabilistic model to predict frames for unseen targets (Das et al., 2010a).[1] Relatedly, for the argument identification subtask, Matsubayashi et al. (2009) proposed a technique for generalization of semantic roles to overcome data sparseness. Unseen targets continue to present a major obstacle to domain-general semantic analysis.

In this paper, we address the problem of idenfifying the semantic frames for targets unseen either in FrameNet (including the exemplar sentences) or the collection of full-text annotations released along with the lexicon. Using a standard model for the argument identification stage (Das et al., 2010a), our proposed method improves overall frame-semantic parsing, especially for unseen targets. To better handle these unseen targets, we adopt a graph-based semi-supervised learning stategy (§4). We construct a large graph over potential targets, most of which

[1] Notwithstanding state-of-the-art results, that approach was only able to identify the correct frame for 1.9% of unseen targets in the test data available at that time. That system achieves about 23% on the test set used in this paper.

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 1435–1444,
Portland, Oregon, June 19-24, 2011.

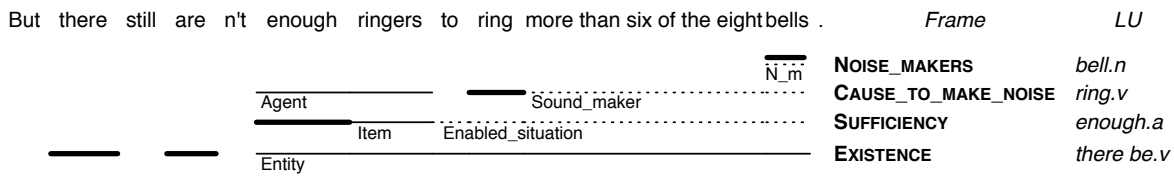

Figure 1: An example sentence from the PropBank section of the full-text annotations released as part of FrameNet 1.5. Each row under the sentence correponds to a semantic frame and its set of corresponding arguments. Thick lines indicate targets that evoke frames; thin solid/dotted lines with labels indicate arguments. N_m under "bells" is short for the Noise_maker role of the NOISE_MAKERS frame.

are drawn from unannotated data, and a fraction of which come from seen FrameNet annotations. Next, we perform label propagation on the graph, which is initialized by frame distributions over the *seen* targets. The resulting smoothed graph consists of posterior distributions over semantic frames for each target in the graph, thus increasing coverage. These distributions are then evaluated within a frame-semantic parser (§5). Considering unseen targets in test data (although few because the test data is also drawn from the training domain), significant absolute improvements of 15.7% and 13.7% are observed for frame identification and full frame-semantic parsing, respectively, indicating improved coverage for hitherto unobserved predicates (§6).

2 Background

Before going into the details of our model, we provide some background on two topics relevant to this paper: frame-semantic parsing and graph-based learning applied to natural language tasks.

2.1 Frame-semantic Parsing

Gildea and Jurafsky (2002) pioneered SRL, and since then there has been much applied research on predicate-argument semantics. Early work on frame-semantic role labeling made use of the exemplar sentences in the FrameNet corpus, each of which is annotated for a single frame and its arguments (Thompson et al., 2003; Fleischman et al., 2003; Shi and Mihalcea, 2004; Erk and Padó, 2006, *inter alia*). Most of this work was done on an older, smaller version of FrameNet. Recently, since the release of full-text annotations in SemEval'07 (Baker et al., 2007), there has been work on identifying multiple frames and their corresponding sets of arguments in a sentence. The LTH system of Johansson and Nugues (2007) performed the best in the SemEval'07 shared task on frame-semantic parsing. Our probabilistic frame-semantic parser outperforms LTH on that task and dataset (Das et al., 2010a). The current paper builds on those probabilistic models to improve coverage on unseen predicates.[2]

Expert resources have limited coverage, and FrameNet is no exception. Automatic induction of semantic resources has been a major effort in recent years (Snow et al., 2006; Ponzetto and Strube, 2007, *inter alia*). In the domain of frame semantics, previous work has sought to extend the coverage of FrameNet by exploiting resources like VerbNet, WordNet, or 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). Although these approaches have increased coverage to various degrees, they rely on other lexicons and resources created by experts. Fürstenau and Lapata (2009) proposed the use of unlabeled data to improve coverage, but their work was limited to verbs. Bejan (2009) used self-training to improve frame identification and reported improvements, but did not explicitly model unknown targets. In contrast, we use statistics gathered from large volumes of unlabeled data to improve the coverage of a frame-semantic parser on several syntactic categories, in a novel framework that makes use of graph-based semi-supervised learning.

[2]SEMAFOR, the system presented by Das et al. (2010a) is publicly available at `http://www.ark.cs.cmu.edu/SEMAFOR` and has been extended in this work.

2.2 Graph-based Semi-Supervised Learning

In graph-based semi-supervised learning, one constructs a graph whose vertices are labeled and unlabeled examples. Weighted edges in the graph, connecting pairs of examples/vertices, encode the degree to which they are expected to have the same label (Zhu et al., 2003). Variants of label propagation are used to transfer labels from the labeled to the unlabeled examples. There are several instances of the use of graph-based methods for natural language tasks. Most relevant to our work an approach to word-sense disambiguation due to Niu et al. (2005). Their formulation was transductive, so that the test data was part of the constructed graph, and they did not consider predicate-argument analysis. In contrast, we make use of the smoothed graph during inference in a probabilistic setting, in turn using it for the full frame-semantic parsing task. Recently, Subramanya et al. (2010) proposed the use of a graph over substructures of an underlying sequence model, and used a smoothed graph for domain adaptation of part-of-speech taggers. Subramanya et al.'s model was extended by Das and Petrov (2011) to induce part-of-speech dictionaries for unsupervised learning of taggers. Our semi-supervised learning setting is similar to these two lines of work and, like them, we use the graph to arrive at better final structures, in an inductive setting (i.e., where a parametric model is learned and then separately applied to test data, following most NLP research).

3 Approach Overview

Our overall approach to handling unobserved targets consists of four distinct stages. Before going into the details of each stage individually, we provide their overview here:

Graph Construction: A graph consisting of vertices corresponding to targets is constructed using a combination of frame similarity (for observed targets) and distributional similarity as edge weights. This stage also determines a fixed set of nearest neighbors for each vertex in the graph.

Label Propagation: The observed targets (a small subset of the vertices) are initialized with empirical frame distributions extracted from FrameNet annotations. Label propagation results in a distribution of frames for each vertex in the graph.

Supervised Learning: Frame identification and argument identification models are trained following Das et al. (2010a). The graph is used to define the set of candidate frames for unseen targets.

Parsing: The frame identification model of Das et al. disambiguated among only those frames associated with a seen target in the annotated data. For an unseen target, all frames in the FrameNet lexicon were considered (a large number). The current work replaces that strategy, considering only the top M frames in the distribution produced by label propagation. This strategy results in large improvements in frame identification for the unseen targets and makes inference much faster. Argument identification is done exactly like Das et al. (2010a).

4 Semi-Supervised Learning

We perform semi-supervised learning by constructing a graph of vertices representing a large number of targets, and learn frame distributions for those which were not observed in FrameNet annotations.

4.1 Graph Construction

We construct a graph with targets as vertices. For us, each target corresponds to a lemmatized word or phrase appended with a coarse POS tag, and it resembles the *lexical units* in the FrameNet lexicon. For example, two targets corresponding to the same lemma would look like *boast*.N and *boast*.V. Here, the first target is a noun, while the second is a verb. An example multiword target is *chemical weapon*.N.

We use two resources for graph construction. First, we take all the words and phrases present in the dependency-based thesaurus constructed using syntactic cooccurrence statistics (Lin, 1998).[3] To construct this resource, a corpus containing 64 million words was parsed with a fast dependency parser (Lin, 1993; Lin, 1994), and syntactic contexts were used to find similar lexical items for a given word

[3]This resource is available at `http://webdocs.cs.ualberta.ca/~lindek/Downloads/sim.tgz`

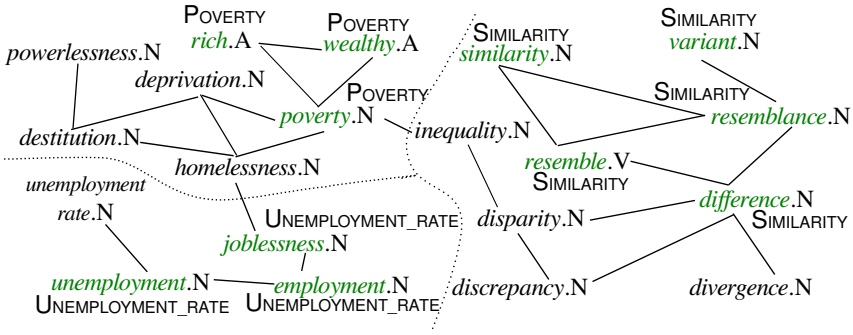

Figure 2: Excerpt from a graph over targets. Green targets are observed in the FrameNet data. Above/below them are shown the most frequently observed frame that these targets evoke. The black targets are unobserved and label propagation produces a distribution over most likely frames that they could evoke.

or phrase. Lin separately treated nouns, verbs and adjectives/adverbs and the thesaurus contains three parts for each of these categories. For each item in the thesaurus, 200 nearest neighbors are listed with a symmetric similarity score between 0 and 1. We processed this thesaurus in two ways: first, we lowercased and lemmatized each word/phrase and merged entries which shared the same lemma; second, we separated the adjectives and adverbs into two lists from Lin's original list by scanning a POS-tagged version of the Gigaword corpus (Graff, 2003) and categorizing each item into an adjective or an adverb depending on which category the item associated with more often in the data. The second step was necessary because FrameNet treats adjectives and adverbs separately. At the end of this processing step, we were left with 61,702 units—approximately six times more than the targets found in FrameNet annotations—each labeled with one of 4 coarse tags. We considered only the top 20 most similar targets for each target, and noted Lin's similarity between two targets t and u, which we call $\mathsf{sim}_{DL}(t, u)$.

The second component of graph construction comes from FrameNet itself. We scanned the exemplar sentences in FrameNet 1.5[4] and the training section of the full-text annotations that we use to train the probabilistic frame parser (see §6.1), and gathered a distribution over frames for each target. For a pair of targets t and u, we measured the Euclidean distance[5] between their frame distributions. This distance was next converted to a similarity score, namely, $\mathsf{sim}_{FN}(t, u)$ between 0 and 1 by subtracting each one from the maximum distance found in the whole data, followed by normalization. Like $\mathsf{sim}_{DL}(t, u)$, this score is symmetric. This resulted in 9,263 targets, and again for each, we considered the 20 most similar targets. Finally, the overall similarity between two given targets t and u was computed as:

$$\mathsf{sim}(t, u) = \alpha \cdot \mathsf{sim}_{FN}(t, u) + (1 - \alpha) \cdot \mathsf{sim}_{DL}(t, u)$$

Note that this score is symmetric because its two components are symmetric. The intuition behind taking a linear combination of the two types of similarity functions is as follows. We hope that distributionally similar targets would have the same semantic frames because ideally, lexical units evoking the same set of frames appear in similar syntactic contexts. We would also like to involve the annotated data in graph construction so that it can eliminate some noise in the automatically constructed thesaurus.[6] Let $\mathcal{K}(t)$ denote the K most similar targets to target t, under the score sim. We link vertices t and u in the graph with edge weight w_{tu}, defined as:

$$w_{tu} = \begin{cases} \mathsf{sim}(t, u) & \text{if } t \in \mathcal{K}(u) \text{ or } u \in \mathcal{K}(t) \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The hyperparameters α and K are tuned by cross-validation (§6.3).

4.2 Label Propagation

First, we softly label those vertices of the constructed graph for which frame distributions are available from the FrameNet data (the same distributions that are used to compute sim_{FN}). Thus, initially, a small fraction of the vertices in the graph

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

[5] This could have been replaced by an entropic distance metric like KL- or JS-divergence, but we leave that exploration to future work.

[6] In future work, one might consider *learning* a similarity metric from the annotated data, so as to exactly suit the frame identification task.

have soft frame labels on them. Figure 2 shows an excerpt from a constructed graph. For simplicity, only the most probable frames under the empirical distribution for the observed targets are shown; we actually label each vertex with the full empirical distribution over frames for the corresponding observed target in the data. The dotted lines demarcate parts of the graph that associate with different frames. Label propagation helps propagate the initial soft labels throughout the graph. To this end, we use a variant of the quadratic cost criterion of Bengio et al. (2006), also used by Subramanya et al. (2010) and Das and Petrov (2011).[7]

Let V denote the set of all vertices in the graph, $V_l \subset V$ be the set of known targets and $\mathcal{F}$ denote the set of all frames. Let $\mathcal{N}(t)$ denote the set of neighbors of vertex $t \in V$. Let $\mathrm{q} = \{\mathrm{q}_1, \mathrm{q}_2, \ldots, \mathrm{q}_{|V|}\}$ be the set of frame distributions, one per vertex. For each known target $t \in V_l$, we have an initial frame distribution r_t. For every edge in the graph, weights are defined as in Eq. 1. We find q by solving:

$$\begin{aligned} \arg\min_{\mathrm{q}} \quad & \textstyle\sum_{t \in V_l} \|\mathrm{r}_t - \mathrm{q}_t\|^2 \\ & + \mu \textstyle\sum_{t \in V, u \in \mathcal{N}(t)} w_{tu} \|\mathrm{q}_t - \mathrm{q}_u\|^2 \\ & + \nu \textstyle\sum_{t \in V} \|\mathrm{q}_t - \frac{1}{|\mathcal{F}|}\|^2 \\ \text{s.t.} \quad & \forall t \in V, \ \textstyle\sum_{f \in \mathcal{F}} \mathrm{q}_t(f) = 1 \\ & \forall t \in V, f \in \mathcal{F}, \ \mathrm{q}_t(f) \geq 0 \end{aligned} \tag{2}$$

We use a squared loss to penalize various pairs of distributions over frames: $\|\mathrm{a} - \mathrm{b}\|^2 = \sum_{f \in \mathcal{F}} (\mathrm{a}(f) - \mathrm{b}(f))^2$. The first term in Eq. 2 requires that, for known targets, we stay close to the initial frame distributions. The second term is the graph smoothness regularizer, which encourages the distributions of similar nodes (large w_{tu}) to be similar. The final term is a regularizer encouraging all distributions to be uniform to the extent allowed by the first two terms. (If an unlabeled vertex does not have a path to any labeled vertex, this term ensures that its converged marginal will be uniform over all frames.) μ and ν are hyperparameters whose choice we discuss in §6.3.

Note that Eq. 2 is convex in q. While it is possible to derive a closed form solution for this objective function, it would require the inversion of a $|V| \times |V|$ matrix. Hence, like Subramanya et al. (2010), we employ an iterative method with updates defined as:

$$\begin{aligned} \gamma_t(f) \quad &\leftarrow \quad \mathrm{r}_t(f)\mathbf{1}\{t \in V_l\} \\ &\quad + \ \mu \sum_{u \in \mathcal{N}(t)} w_{tu} \mathrm{q}_u^{(m-1)}(f) + \frac{\nu}{|\mathcal{F}|} \end{aligned} \tag{3}$$

$$\kappa_t \quad \leftarrow \quad \mathbf{1}\{t \in V_l\} + \nu + \mu \sum_{u \in \mathcal{N}(t)} w_{tu} \tag{4}$$

$$\mathrm{q}_t^{(m)}(f) \quad \leftarrow \quad \gamma_t(f)/\kappa_t \tag{5}$$

Here, $\mathbf{1}\{\cdot\}$ is an indicator function. The iterative procedure starts with a uniform distribution for each $\mathrm{q}_t^{(0)}$. For all our experiments, we run 10 iterations of the updates. The final distribution of frames for a target t is denoted by q_t^*.

5 Learning and Inference for Frame-Semantic Parsing

In this section, we briefly review learning and inference techniques used in the frame-semantic parser, which are largely similar to Das et al. (2010a), except the handling of unknown targets. Note that in all our experiments, we assume that the targets are marked in a given sentence of which we want to extract a frame-semantic analysis. Therefore, unlike the systems presented in SemEval'07, we do not define a target identification module.

5.1 Frame Identification

For a given sentence $\mathbf{x}$ with frame-evoking targets $\mathbf{t}$, let t_i denote the ith target (a word sequence). We seek a list $\mathbf{f} = \langle f_1, \ldots, f_m \rangle$ of frames, one per target. Let $\mathcal{L}$ be the set of targets found in the FrameNet annotations. Let $\mathcal{L}_f \subseteq \mathcal{L}$ be the subset of these targets annotated as evoking a particular frame f.

The set of candidate frames $\mathcal{F}_i$ for t_i is defined to include every frame f such that $t_i \in \mathcal{L}_f$. If $t_i \notin \mathcal{L}$ (in other words, t_i is unseen), then Das et al. (2010a) considered all frames $\mathcal{F}$ in FrameNet as candidates. Instead, in our work, we check whether $t_i \in V$, where V are the vertices of the constructed graph, and set:

$$\mathcal{F}_i = \{f : f \in M\text{-best frames under } \mathrm{q}_{t_i}^*\} \tag{6}$$

The integer M is set using cross-validation (§6.3). If $t_i \notin V$, then all frames $\mathcal{F}$ are considered as $\mathcal{F}_i$.

[7] Instead of a quadratic cost, an entropic distance measure could have been used, e.g., KL-divergence, considered by Subramanya and Bilmes (2009). We do not explore that direction in the current paper.

The frame prediction rule uses a probabilistic model over frames for a target:

$$f_i \leftarrow \arg\max_{f \in \mathcal{F}_i} \sum_{\ell \in \mathcal{L}_f} p(f, \ell \mid t_i, \mathbf{x}) \quad (7)$$

Note that a latent variable $\ell \in \mathcal{L}_f$ is used, which is marginalized out. Broadly, lexical semantic relationships between the "prototype" variable ℓ (belonging to the set of seen targets for a frame f) and the target t_i are used as features for frame identification, but since ℓ is unobserved, it is summed out both during inference and training. A conditional log-linear model is used to model this probability: for $f \in \mathcal{F}_i$ and $\ell \in \mathcal{L}_f$, $p_{\boldsymbol{\theta}}(f, \ell \mid t_i, \mathbf{x}) =$

$$\frac{\exp \boldsymbol{\theta}^\top \mathbf{g}(f, \ell, t_i, \mathbf{x})}{\sum_{f' \in \mathcal{F}_i} \sum_{\ell' \in \mathcal{L}_{f'}} \exp \boldsymbol{\theta}^\top \mathbf{g}(f', \ell', t_i, \mathbf{x})} \quad (8)$$

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 f to a prototype ℓ, represent a lexical-semantic relationship between ℓ and t_i, or encode part of the syntax of the sentence (Das et al., 2010b).

Given some training data, which is of the form $\left\langle \langle \mathbf{x}^{(j)}, \mathbf{t}^{(j)}, \mathbf{f}^{(j)}, \mathcal{A}^{(j)} \rangle \right\rangle_{j=1}^{N}$ (where N is the number of sentences in the data and $\mathcal{A}$ is the set of argument in a sentence), we discriminatively train the frame identification model by maximizing the following log-likelihood:[8]

$$\max_{\boldsymbol{\theta}} \sum_{j=1}^{N} \sum_{i=1}^{m_j} \log \sum_{\ell \in \mathcal{L}_{f_i^{(j)}}} p_{\boldsymbol{\theta}}(f_i^{(j)}, \ell \mid t_i^{(j)}, \mathbf{x}^{(j)}) \quad (9)$$

This non-convex objective function is locally optimized using a distributed implementation of L-BFGS (Liu and Nocedal, 1989).[9]

5.2 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 or SRL is the task of choosing which of each f_i's roles are filled, and by which parts of $\mathbf{x}$. We directly adopt the model of Das et al. (2010a) for the argument identification stage and briefly describe it here.

Let $\mathcal{R}_{f_i} = \{r_1, \ldots, r_{|\mathcal{R}_{f_i}|}\}$ denote frame f_i's **roles** observed in FrameNet annotations. A set $\mathcal{S}$ of spans that are candidates for filling any role $r \in \mathcal{R}_{f_i}$ are identified in the sentence. In principle, $\mathcal{S}$ could contain any subsequence of $\mathbf{x}$, but we consider only 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 a dependency parse. The empty span is also included in $\mathcal{S}$, since some roles are not explicitly filled. During training, if an argument is not a valid subtree of the dependency parse (this happens due to parse errors), we add its span to $\mathcal{S}$. Let $\mathcal{A}_i$ denote the mapping of roles in $\mathcal{R}_{f_i}$ to spans in $\mathcal{S}$. The model makes a prediction for each $\mathcal{A}_i(r_k)$ (for all roles $r_k \in \mathcal{R}_{f_i}$):

$$\mathcal{A}_i(r_k) \leftarrow \arg\max_{s \in \mathcal{S}} p(s \mid r_k, f_i, t_i, \mathbf{x}) \quad (10)$$

A conditional log-linear model over spans for each role of each evoked frame is defined as:

$$p_{\boldsymbol{\psi}}(\mathcal{A}_i(r_k) = s \mid f_i, t_i, \mathbf{x}) = \frac{\exp \boldsymbol{\psi}^\top \mathbf{h}(s, r_k, f_i, t_i, \mathbf{x})}{\sum_{s' \in \mathcal{S}} \exp \boldsymbol{\psi}^\top \mathbf{h}(s', r_k, f_i, t_i, \mathbf{x})} \quad (11)$$

This model is trained by optimizing:

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

This objective function is convex, and we globally optimize it using the distributed implementation of L-BFGS. We regularize by including $-\frac{1}{10}\|\boldsymbol{\psi}\|_2^2$ in the objective (the strength is not tuned). Naïve prediction of roles using Equation 10 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. Hence, illegal overlap is disallowed using a 10,000-hypothesis beam search.

[8] We found no benefit from using an L_2 regularizer.

[9] While training, in the partition function of the log-linear model, all frames $\mathcal{F}$ in FrameNet are summed up for a target t_i instead of only $\mathcal{F}_i$ (as in Eq. 8), to learn interactions between the latent variables and different sentential contexts.

Model	Unknown Targets		All Targets	
	Exact Match	**Partial Match**	**Exact Match**	**Partial Match**
SEMAFOR	23.08	46.62	82.97	90.51
Self-training	18.88	42.67	82.45	90.19
LinGraph	**36.36**	**59.47**	**83.40**	**90.93**
FullGraph	**39.86**	**62.35***	**83.51**	**91.02***

Table 1: Frame identification results in percentage accuracy on 4,458 test targets. Bold scores indicate significant improvements relative to SEMAFOR and (*) denotes significant improvements over LinGraph ($p < 0.05$).

6 Experiments and Results

Before presenting our experiments and results, we will describe the datasets used in our experiments, and the various baseline models considered.

6.1 Data

We make use of the FrameNet 1.5 lexicon released in 2010. This lexicon is a superset of previous versions of FrameNet. It contains 154,607 exemplar sentences with one marked target and frame-role annotations. 78 documents with full-text annotations with multiple frames per sentence were also released (a superset of the SemEval'07 dataset). We randomly selected 55 of these documents for training and treated the 23 remaining ones as our test set. After scanning the exemplar sentences and the training data, we arrived at a set of 877 frames, 1,068 roles,[10] and 9,263 targets. Our training split of the full-text annotations contained 3,256 sentences with 19,582 frame annotatations with corresponding roles, while the test set contained 2,420 sentences with 4,458 annotations (the test set contained fewer annotated targets per sentence). We also divide the 55 training documents into 5 parts for cross-validation (see §6.3). The raw sentences in all the training and test documents were preprocessed using MXPOST (Ratnaparkhi, 1996) and the MST dependency parser (McDonald et al., 2005) following Das et al. (2010a). In this work we assume the frame-evoking targets have been correctly identified in training and test data.

[10]Note that the number of listed roles in the lexicon is nearly 9,000, but their number in actual annotations is a lot fewer.

6.2 Baselines

We compare our model with three baselines. The first baseline is the purely supervised model of Das et al. (2010a) trained on the training split of 55 documents. Note that this is the strongest baseline available for this task;[11] we refer to this model as "SEMAFOR."

The second baseline is a semi-supervised self-trained system, where we used SEMAFOR to label 70,000 sentences from the Gigaword corpus with frame-semantic parses. For finding targets in a raw sentence, we used a relaxed target identification scheme, where we marked every target seen in the lexicon and all other words which were not prepositions, particles, proper nouns, foreign words and Wh-words as potential frame evoking units. This was done so as to find unseen targets and get frame annotations with SEMAFOR on them. We appended these automatic annotations to the training data, resulting in 711,401 frame annotations, more than 36 times the supervised data. These data were next used to train a frame identification model (§5.1).[12] This setup is very similar to Bejan (2009) who used self-training to improve frame identification. We refer to this model as "Self-training."

The third baseline uses a graph constructed only with Lin's thesaurus, without using supervised data. In other words, we followed the same scheme as in §4.1 but with the hyperparameter $\alpha = 0$. Next, label propagation was run on this graph (and hyperparameters tuned using cross validation). The posterior distribution of frames over targets was next used for frame identification (Eq. 6-7), with SEMAFOR as the trained model. This model, which is very similar to our full model, is referred to as "LinGraph."

"FullGraph" refers to our full system.

6.3 Experimental Setup

We used five-fold cross-validation to tune the hyperparameters α, K, μ, and M in our model. The

[11]We do not compare our model with other systems, e.g. the ones submitted to SemEval'07 shared task, because SEMAFOR outperforms them significantly (Das et al., 2010a) on the previous version of the data. Moreover, we trained our models on the new FrameNet 1.5 data, and training code for the SemEval'07 systems was not readily available.

[12]Note that we only self-train the frame identification model and not the argument identification model, which is fixed throughout.

Model	UNKNOWN TARGETS						ALL TARGETS					
	Exact Match			Partial Match			Exact Match			Partial Match		
	P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
SEMAFOR	19.59	16.48	17.90	33.03	27.80	30.19	66.15	61.64	63.82	70.68	65.86	68.18
Self-training	15.44	13.00	14.11	29.08	24.47	26.58	65.78	61.30	63.46	70.39	65.59	67.90
LinGraph	**29.74**	**24.88**	**27.09**	**44.08**	**36.88**	**40.16**	**66.43**	**61.89**	**64.08**	**70.97**	**66.13**	**68.46**
FullGraph	**35.27***	**28.84***	**31.74***	**48.81***	**39.91***	**43.92***	**66.59***	**62.01***	**64.22***	**71.11***	**66.22***	**68.58***

Table 2: Full frame-semantic parsing precision, recall and F_1 score on 2,420 test sentences. Bold scores indicate significant improvements relative to SEMAFOR and (*) denotes significant improvements over LinGraph ($p < 0.05$).

uniform regularization hyperparameter ν for graph construction was set to 10^{-6} and not tuned. For each cross-validation split, four folds were used to train a frame identification model, construct a graph, run label propagation and then the model was tested on the fifth fold. This was done for all hyperparameter settings, which were $\alpha \in \{0.2, 0.5, 0.8\}$, $K \in \{5, 10, 15, 20\}$, $\mu \in \{0.01, 0.1, 0.3, 0.5, 1.0\}$ and $M \in \{2, 3, 5, 10\}$. The joint setting which performed the best across five-folds was $\alpha = 0.2, K = 10, \mu = 1.0, M = 2$. Similar tuning was also done for the baseline LinGraph, where α was set to 0, and rest of the hyperparameters were tuned (the selected hyperparameters were $K = 10$, $\mu = 0.1$ and $M = 2$). With the chosen set of hyperparameters, the test set was used to measure final performance.

The standard evaluation script from the SemEval'07 task calculates precision, recall, and F_1-score for frames and arguments; it also provides a score that gives partial credit for hypothesizing a frame *related* to the correct one in the FrameNet lexicon. 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. This evaluation scheme follows Das et al. (2010a). Statistical significance is measured using a reimplementation of Dan Bikel's parsing evaluation comparator.[13]

6.4 Results

Tables 1 and 2 present results for frame identification and full frame-semantic parsing respectively. They also separately tabulate the results achieved for unknown targets. Our full model, denoted by "FullGraph," outperforms all the baselines for both tasks. Note that the Self-training model even falls short of the supervised baseline SEMAFOR, unlike what was observed by Bejan (2009) for the frame identification task. The model using a graph constructed solely from the thesaurus (LinGraph) outperforms both the supervised and the self-training baselines for all tasks, but falls short of the graph constructed using the similarity metric that is a linear combination of distributional similarity and supervised frame similarity. This indicates that a graph constructed with some knowledge of the supervised data is more powerful.

For unknown targets, the gains of our approach are impressive: 15.7% absolute accuracy improvement over SEMAFOR for frame identification, and 13.7% absolute F_1 improvement over SEMAFOR for full frame-semantic parsing (both significant). When all the test targets are considered, the gains are still significant, resulting in 5.4% relative error reduction over SEMAFOR for frame identification, and 1.3% relative error reduction over SEMAFOR for full-frame semantic parsing.

Although these improvements may seem modest, this is because only 3.2% of the test set targets are unseen in training. We expect that further gains would be realized in different text domains, where FrameNet coverage is presumably weaker than in news data. A semi-supervised strategy like ours is attractive in such a setting, and future work might explore such an application.

Our approach also makes decoding much faster. For the unknown component of the test set, SEMAFOR takes a total 111 seconds to find the best set of frames, while the FullGraph model takes only 19 seconds to do so, thus bringing disambiguation time down by a factor of nearly 6. This is because our model now disambiguates between only $M = 2$ frames instead of the full set of 877 frames in FrameNet. For the full test set too, the speedup

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

$t =$ *discrepancy*.N		$t =$ *contribution*.N		$t =$ *print*.V		$t =$ *mislead*.V	
f	$\mathrm{q}_t^*(f)$	f	$\mathrm{q}_t^*(f)$	f	$\mathrm{q}_t^*(f)$	f	$\mathrm{q}_t^*(f)$
∗SIMILARITY	0.076	∗GIVING	0.167	∗TEXT_CREATION	0.081	EXPERIENCER_OBJ	0.152
NATURAL_FEATURES	0.066	MONEY	0.046	SENDING	0.054	∗PREVARICATION	0.130
PREVARICATION	0.012	COMMITMENT	0.046	DISPERSAL	0.054	MANIPULATE_INTO_DOING	0.046
QUARRELING	0.007	ASSISTANCE	0.040	READING	0.042	COMPLIANCE	0.041
DUPLICATION	0.007	EARNINGS_AND_LOSSES	0.024	STATEMENT	0.028	EVIDENCE	0.038

Table 3: Top 5 frames according to the graph posterior distribution $\mathrm{q}_t^*(f)$ for four targets: *discrepancy*.N, *contribution*.N, *print*.V and *mislead*.V. None of these targets were present in the supervised FrameNet data. $*$ marks the correct frame, according to the test data. EXPERIENCER_OBJ is described in FrameNet as "Some phenomenon (the Stimulus) provokes a particular emotion in an Experiencer."

is noticeable, as SEMAFOR takes 131 seconds for frame identification, while the FullGraph model only takes 39 seconds.

6.5 Discussion

The following is an example from our test set showing SEMAFOR's output (for one target):

> REASON
> Discrepancies between North Korean declarations and IAEA inspection findings Action indicate that North Korea might have reprocessed enough plutonium for one or two nuclear weapons.
> *discrepancy*.N

Note that the model identifies an incorrect frame REASON for the target *discrepancy*.N, in turn identifying the wrong semantic role Action for the underlined argument. On the other hand, the FullGraph model exactly identifies the right semantic frame, SIMILARITY, as well as the correct role, Entities. This improvement can be easily explained. The excerpt from our constructed graph in Figure 2 shows the same target *discrepancy*.N in black, conveying that it did not belong to the supervised data. However, it is connected to the target *difference*.N drawn from annotated data, which evokes the frame SIMILARITY. Thus, after label propagation, we expect the frame SIMILARITY to receive high probability for the target *discrepancy*.N.

Table 3 shows the top 5 frames that are assigned the highest posterior probabilities in the distribution q_t^* for four hand-selected test targets absent in supervised data, including *discrepancy*.N. For all of them, the FullGraph model identifies the correct frames for all four words in the test data by ranking these frames in the top $M = 2$. LinGraph also gets all four correct, Self-training only gets *print*.V/TEXT_CREATION, and SEMAFOR gets none.

Across unknown targets, on average the $M = 2$ most common frames in the posterior distribution q_t^* found by FullGraph have $\mathrm{q}_t^{(*)}(f) = \frac{7}{877}$, or seven times the average across all frames. This suggests that the graph propagation method is confident only in predicting the top few frames out of the whole possible set. Moreover, the automatically selected number of frames to extract per unknown target, $M = 2$, suggests that only a few meaningful frames were assigned to unknown predicates. This matches the nature of FrameNet data, where the average frame ambiguity for a target type is 1.20.

7 Conclusion

We have presented a semi-supervised strategy to improve the coverage of a frame-semantic parsing model. We showed that graph-based label propagation and resulting smoothed frame distributions over unseen targets significantly improved the coverage of a state-of-the-art semantic frame disambiguation model to previously unseen predicates, also improving the quality of full frame-semantic parses. The improved parser is available at `http://www.ark.cs.cmu.edu/SEMAFOR`.

Acknowledgments

We are grateful to Amarnag Subramanya for helpful discussions. We also thank Slav Petrov, Nathan Schneider, and the three anonymous reviewers for valuable comments. This research was supported by NSF grants IIS-0844507, IIS-0915187 and TeraGrid resources provided by the Pittsburgh Supercomputing Center under NSF grant number TG-DBS110003.

References

C. Baker, M. Ellsworth, and K. Erk. 2007. SemEval-2007 Task 19: frame semantic structure extraction. In *Proc. of SemEval*.

C. A. Bejan. 2009. *Learning Event Structures From Text*. Ph.D. thesis, The University of Texas at Dallas.

Y. Bengio, O. Delalleau, and N. Le Roux. 2006. Label propagation and quadratic criterion. In *Semi-Supervised Learning*. MIT Press.

H. C. Boas. 2002. Bilingual FrameNet dictionaries for machine translation. In *Proc. of LREC*.

D. Das and S. Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proc. of ACL-HLT*.

D. Das, N. Schneider, D. Chen, and N. A. Smith. 2010a. Probabilistic frame-semantic parsing. In *Proc. of NAACL-HLT*.

D. Das, N. Schneider, D. Chen, and N. A. Smith. 2010b. SEMAFOR 1.0: A probabilistic frame-semantic parser. Technical Report CMU-LTI-10-001, Carnegie Mellon University.

K. Erk and S. Padó. 2006. Shalmaneser - a toolchain for shallow semantic parsing. In *Proc. of LREC*.

C. Fellbaum, editor. 1998. *WordNet: an electronic lexical database*. MIT Press, Cambridge, MA.

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

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

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

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

H. Fürstenau and M. Lapata. 2009. Semi-supervised semantic role labeling. In *Proc. of EACL*.

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

A.-M. Giuglea and A. Moschitti. 2006. Shallow semantic parsing based on FrameNet, VerbNet and PropBank. In *Proc. of ECAI 2006*.

D. Graff. 2003. English Gigaword. Linguistic Data Consortium.

R. Johansson and P. Nugues. 2007. LTH: semantic structure extraction using nonprojective dependency trees. In *Proc. of SemEval*.

D. Lin. 1993. Principle-based parsing without overgeneration. In *Proc. of ACL*.

D. Lin. 1994. Principar–an efficient, broadcoverage, principle-based parser. In *Proc. of COLING*.

D. Lin. 1998. Automatic retrieval and clustering of similar words. In *Proc. of COLING-ACL*.

D. C. Liu and J. Nocedal. 1989. On the limited memory bfgs method for large scale optimization. *Math. Programming*, 45(3).

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

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

Z.-Y. Niu, D.-H. Ji, and C. L. Tan. 2005. Word sense disambiguation using label propagation based semi-supervised learning. In *Proc. of ACL*.

S. Padó and M. Lapata. 2005. Cross-linguistic projection of role-semantic information. In *Proc. of HLT-EMNLP*.

M. Pennacchiotti, D. De Cao, R. Basili, D. Croce, and M. Roth. 2008. Automatic induction of FrameNet lexical units. In *Proc. of EMNLP*.

S. P. Ponzetto and M. Strube. 2007. Deriving a large scale taxonomy from wikipedia. In *Proc. of AAAI*.

A. Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proc. of EMNLP*.

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

L. Shi and R. Mihalcea. 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.

R. Snow, D. Jurafsky, and A. Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proc. of COLING-ACL*.

A. Subramanya and J. A. Bilmes. 2009. Entropic graph regularization in non-parametric semi-supervised classification. In *Proc. of NIPS*.

A. Subramanya, S. Petrov, and F. Pereira. 2010. Efficient Graph-based Semi-Supervised Learning of Structured Tagging Models. In *Proc. of EMNLP*.

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

S. Tonelli and C. Giuliano. 2009. Wikipedia as frame information repository. In *Proc. of EMNLP*.

X. Zhu, Z. Ghahramani, and J. D. Lafferty. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proc. of ICML*.