Evaluation of Semantic Dependency Labeling Across Domains

Svetlana Stoyanchev
Interactions Labs
25 Broadway,
New York, NY
sstoyanchev@interactions.com

Amanda Stent
Yahoo Labs
229 W. 43rd St.
New York, NY
stent@yahoo-inc.com

Srinivas Bangalore
Interactions Labs
41 Spring Street,
Murray Hill, NJ 07974
sbangalore@interactions.com

Abstract

One of the key concerns in computational semantics is to construct a domain independent semantic representation which captures the richness of natural language, yet can be quickly customized to a specific domain for practical applications. We propose to use generic semantic frames defined in FrameNet, a domain-independent semantic resource, as an intermediate semantic representation for language understanding in dialog systems. In this paper we: (a) outline a novel method for FrameNet-style semantic dependency labeling that builds on a syntactic dependency parse; and (b) compare the accuracy of domain-adapted and generic approaches to semantic parsing for dialog tasks, using a frame-annotated corpus of human-computer dialogs in an airline reservation domain.

Introduction

With advancements in artificial intelligence new types of systems are emerging, for which speech is a natural input modality. Speech is increasingly used with autonomous cars and robots. Speech also eliminates the need for graphical user interfaces in "smart home" systems; users may turn the lights on, change the temperature of a thermostat, choose music to listen to, or control robotic vacuum cleaners using spoken commands. On mobile devices, apps can be controlled through speech; for example, a user can find a restaurant in a search application or retrieve schedule information from a calendar application.

A Spoken Language Understanding (SLU) component for a speech interface identifies *intents* and related *entities* in spoken natural language utterances. For example, the utterance *"Leaving from New York on Friday"* has a *Departing* intent with the arguments location (*New York*) and time (*Friday*). A SLU must be *quick* to support high interactivity, *accurate* for retaining user engagement, and *precise* for mapping into domain-specific functions or database queries. For these reasons SLUs are often domain specific. To create a domain-specific SLU, a system developer annotates domain-specific data or creates rules that convert a user's spoken input into a domain-specific representation (Williams et al. 2015; Wit.AI 2015). Designing a domain-specific SLU is a laborious and time-consuming process. As speech interfaces evolve into general purpose personal assistants, SLUs need to also achieve *generality* and *rapid adaptability* to new domains.

One way to achieve generality and rapid domain adaptation is to use a **generic semantic representation** for SLU with a mapping from generic to application-specific representations in a later control component (Dzikovska, Allen, and Swift 2008; Bastianelli et al. 2014). FrameNet (Fillmore 1982; Baker, Fillmore, and Lowe 1998) is one such representation. FrameNet defines a set of domain-independent and language-independent semantic frames. Each FrameNet frame is associated with a (non-exhaustive) list of *target* words and typed *arguments*. For example, a frame representing the notion of *Departing* is associated with the words *leave, decamp, exit, go,* etc. with the core arguments *source* (location) and *theme* (physical object). FrameNet is not only a lexicon, but also a growing dataset with 170K manually annotated English sentences, providing a unique resource for training SLUs.

There are existing general purpose statistical semantic parsers based on FrameNet annotations (Das et al. 2010; Johansson and Nugues 2007; Croce and Basili 2011). While generic semantic resources have been successfully used for domain-specific SLUs by researchers (Bastianelli et al. 2014; Tur, Hakkani-Tür, and Chotimongkol 2005), they are not widely used in commercial spoken dialog systems due to accuracy issues, some related to the handling of speech disfluencies in spoken language input, and some related to domain-specific coverage gaps.

In this paper, we present a systematic evaluation of approaches to domain adaptation of generic semantic resources for SLU. We define a **semantic dependency labeling** (SDL) task, which is similar to the semantic parsing task (Das et al. 2010) except that frame targets and arguments are labeled on single nodes (phrase heads) in a dependency tree rather than on text spans. Because it uses features associated with a head word of the argument instead of the full argument span, an SDL system should be more robust to speech recognition errors and disfluencies than existing semantic parsers. In addition, a semantically annotated dependency tree provides a rich joint representation of semantic and syntactic information that may be useful for downstream processing.

We present a statistical SDL system, *OnePar*. *OnePar* achieves similar accuracy to the reported results of the widely used state-of-the-art semantic parser SEMAFOR on

FrameNet data. However, our goal is to explore its utility for domain adaptation in SLU. For this purpose we manually annotated the unique sentences in the *Communicator 2000* corpus of air travel dialogs (Walker and others 2001) with FrameNet verb frames[1]. Using this data, we compare the accuracy of: (a) a generic model for *OnePar* trained on FrameNet data; (b) the generic model plus some handwritten domain-specific constraints (such as a dialog system developer would know while designing the system); (c) a model trained on FrameNet data and a small amount of domain-specific training data (such as could be labeled in under a week by trained annotators from a seed corpus for the domain, during dialog system design), plus the domain-specific constraints; and (d) a model trained only on domain-specific training data. We show that domain-specific constraints and a small amount of in-domain training data significantly improve SDL accuracy in-domain, while allowing for rapid bootstrapping of coverage in new domains.

The rest of the paper is organized as follows. First, we describe related work on semantic parsing. Then, we present *OnePar* and the datasets we used in our experiments. Finally, we present and discuss our experimental results.

Related Work

In the field of computational semantics, three semantic resources are widely used for semantic parsing – PropBank (Kingsbury, Palmer, and Marcus 2002), VerbNet (Kipper et al. 2008) and FrameNet (Lowe, Baker, and Fillmore 1997). PropBank is a set of coarse-grained verb frame annotations over the Wall Street Journal Penn Treebank corpus (Marcus, Santorini, and Marcinkiewicz 1993), and was used in the CoNLL 2004 and 2005 semantic role labeling shared tasks (Carreras and Màrquez 2005). VerbNet and FrameNet are both semantic lexicons incorporating fine-grained verb frame specifications, with large sets of labeled examples. FrameNet was used in the CoNLL 2007 semantic parsing shared task (Johansson and Nugues 2007)[2].

Frame-semantic parsing involves identification and disambiguation of the frames associated with certain key words (*targets*) and their *arguments* in a sentence. The most common approaches to FrameNet-based semantic parsing have been supervised (e.g. (Das et al. 2010; Johansson and Nugues 2007)). Recently, unsupervised and semi-supervised approaches have also been investigated (Lang and Lapata 2014; Das et al. 2014); however, these approaches still lack the accuracy required for SLU. As described below, *OnePar* uses a fairly standard three-stage approach with supervised training methods for each stage.

Chen et al. (2013) apply generic FrameNet-based semantic parsing for SLU. However, they only use the target labels, ignoring the arguments. Others have trained frame-semantic SLUs with domain-specific data (Artzi and Zettlemoyer 2011; Coppola et al. 2008; Coppola, Moschitti, and Riccardi 2009). However, the effort required to annotate a sizeable corpus of dialogs for each new domain with FrameNet frames is large.

Bastianelli et al. (2014) use a FrameNet-style SLU for interpreting commands to an autonomous robot. Their method of domain adaptation is to manually select a set of FrameNet frames corresponding to their domain. They use a purpose-built SLU that is tightly integrated with speech recognition.

Tur et al. (2005) use a generic PropBank-style parser for semi-supervised intent identification. This method does not identify arguments. Huang and Yates (2010) present domain adaptation methods for PropBank-style semantic parsing using domain-independent preprocessing techniques. Their method requires only unlabeled domain-specific data. However, there is still a “chicken and egg” problem: in order to collect a large set of dialogs a dialog system (with an SLU) would generally have to be running already.

System: *OnePar*

The architecture for *OnePar* is shown in Figure 1. The input is a syntactically processed sentence. We use our own tools for syntactic processing. SDL is performed in three stages typical of frame semantic parsers: (1) target labeling; (2) argument labeling; and (3) frame fitting.

Data Preprocessing

Targets and arguments in FrameNet data are labeled on spans of words, or chunks. Consequently, most semantic parsers model argument labeling as the tagging of word spans under optimization constraints, such as *no span overlap* and *unique core roles* (Punyakanok, Roth, and Yih 2008; Täckström, Ganchev, and Das 2015). By contrast, *OnePar* labels head nodes in a dependency parse[3], an approach we borrow from the CoNLL shared tasks on PropBank-style semantic role labeling (Surdeanu et al. 2008; Hajič et al. 2009). For example, in the FrameNet annotation of the sentence depicted in Figure 2, *“a clean sheet of paper”* is labeled as an argument span. However, alternative valid argument spans may be *“sheet of paper”* or *“clean sheet”*. In our approach, we defer span identification and instead label head words of frame arguments. Spans can be identified from the dependency tree at a later stage if the application requires. This approach allows for more robust handling of incomplete or error-filled input as it does not require the argument span to be a proper subtree.

To convert FrameNet argument span annotations to argument head word annotations for training and testing, we identify head words of each span in a manner similar to (Bauer, Fürstenau, and Rambow 2012): each sentence is dependency parsed; then, for each annotated argument span, if the span corresponds to a proper dependency constituent (as in the example on Figure 2), we mark its head word; else if

[1] Working with the Linguistic Data Consortium, we plan to release these annotations in the near future.

[2] As Martha Palmer noted in 2009 (http://www.flarenet.eu/sites/default/files/S3_01_Palmer.pdf), “PropBank provides the best training data”, “VerbNet provides the clearest links between syntax and semantics”, and “FrameNet provides the richest semantics”.

[3] A dependency parse is a graph representation of the syntactic structure of a sentence in which each pair of grammatically related words is connected with an edge (see Figure 2).

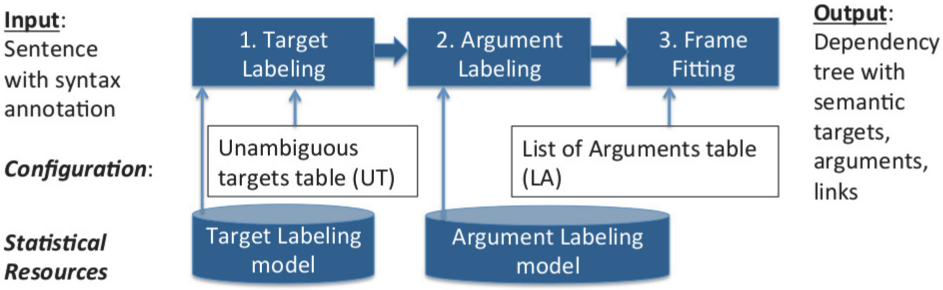


Figure 1: *OnePar* system

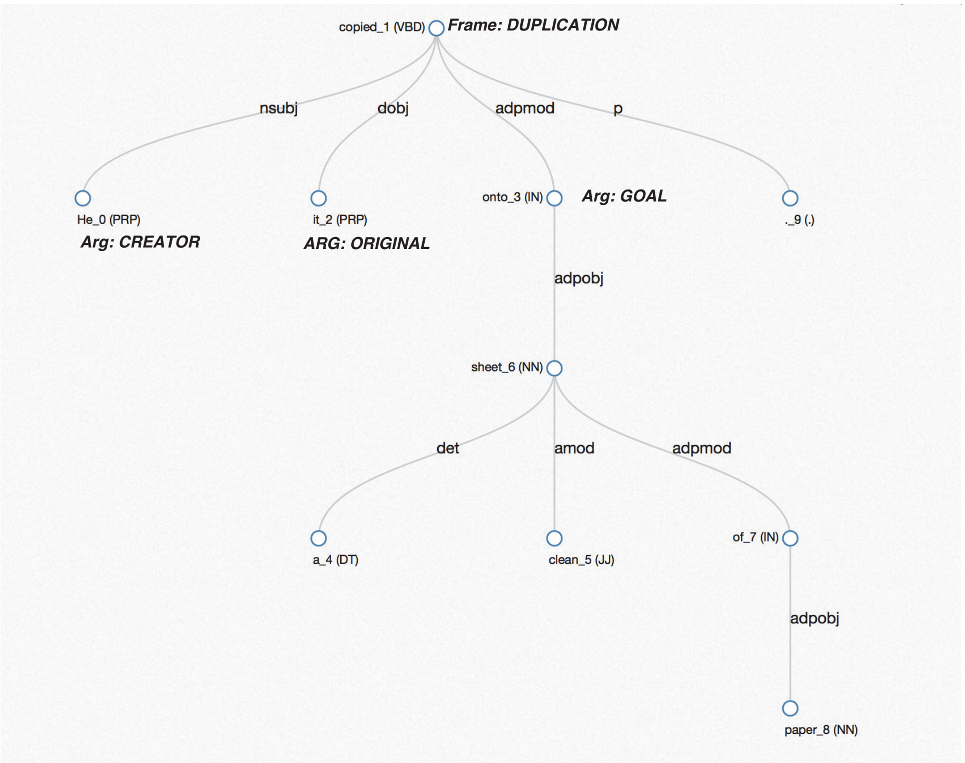


Figure 2: Dependency parse for a sentence *"He copied it onto a clean sheet of paper"* tagged with a frame and arguments.

there is a noun in the argument, we mark the first noun as the head; else we mark the last word as the head [4].

Target Labeling

In the first stage of SDL, we identify and label target words in the sentence. From the set of all FrameNet frames $\mathcal{F}$, target labeling assigns a frame identifier $F_w \in \mathcal{F}$ to each candidate target w, as determined by Equation 1:

$$\begin{aligned} F_w &= UT(w, p_w)\ if\ defined \qquad (1) \\ &= \underset{F}{argmax}\ P(F \mid \phi_s, \phi_d, \phi_c)\ otherwise \end{aligned}$$

where $UT(w, p_w)$ is an *unambiguous targets* table that uniquely identifies a frame given the word (w) and its part-of-speech (POS), p_w[5].

Candidate targets may include all words in the sentence (all nodes in the dependency tree), or only selected words (nodes). If all words in the sentence are considered as targets, target labeling assigns a frame identifier $F_w \in \mathcal{F} \cup \{None\}$ for joint identification and labeling. For SLU, we are interested in verb frames, so in our experiments only verbs are considered as candidate targets. Target labeling in this case assigns a frame identifier $F_w \in \mathcal{F}$ to every verb in a sentence.

[4] Only 20% of arguments in FrameNet data do not correspond to a proper subtree in the parse from our dependency parser.

[5] The majority of target words (86%) in the FrameNet corpus are unambiguously associated with a frame.

Feature Type	Used in	Features List
Sequence (ϕ_s)	TARGET, ARG	word, stem, part of speech tag (POS), word/POS context (2 right, 2 left)
Dependency Parse (ϕ_d)	TARGET, ARG	parent relation, parent word, parent POS, child relation, child POS
Constituency Parse (ϕ_c)	TARGET, ARG	non-terminal (NT) value of the corresponding node in the constituency parse, left/right sibling word/POS/NT, parent node's word/POS/NT
Dependency Parse Link (ϕ_l)	ARG	path of POS/relations between target and candidate argument, length of the path, dependency join word/POS/distance to target (if path between target and candidate argument is indirect)
Constituency Parse Link (ϕ_l)	ARG	path of POS/relations between target and candidate argument, path length

Table 1: Features for target and argument labeling. *Used in* column shows which model is each feature set used in.

$P(F \mid \phi_s, \phi_d, \phi_c)$ is computed by a maximum entropy classifier using the sequence (ϕ_s), dependency tree (ϕ_d), and constituency tree (ϕ_c) features outlined in Table 1. Sequence features include tokens, stems, POS tags and their context. Dependency features include parent-relation and POS information for the candidate target and its parent, and the relation between them. Constituency features include non-terminal and POS information for the candidate target, its left and right siblings, and its parent.

Argument Labeling

The second stage of SDL is argument labeling. For each target labeled in the first stage, the argument labeler determines the probability of each head word in the dependency tree being an argument of the corresponding frame. Each frame F requires a set of arguments A_F. Although according to the FrameNet annotation scheme, each frame is unique, in fact many frames share common argument types, such as *Agent, Entity, Event, Manner, Means, Place* or *Time*. We exploit this redundancy to address the data sparseness problem in FrameNet. We leverage argument annotation data across different frames by creating a single argument label set ($\mathcal{A} = \bigcup_F A_F \cup \{None\}$) and training a single model for argument labeling.

OnePar labels each word (w) in a sentence with a probability distribution over argument labels $P(a \in \mathcal{A} \mid w, \phi_s, \phi_d, \phi_c, \phi_l)$ using a maximum entropy model with features ϕ_s, ϕ_d and ϕ_c, defined as for target identification, and ϕ_l, link features defining the path between target and argument in dependency and constituency parses (see Table 1).

Frame Fitting

Given a target, its associated frame F and arguments A_F, and a probability over potential argument labels for each non-target word, we want to find the most likely assignment for each $a_i \in A_F$. Frame arguments are defined in a *list of arguments (LA)* table[6]. The final assignment of argument labels to words is made according to the following:

$$a_w = \underset{a \in A_F}{argmax} P(a \mid w, \phi_s, \phi_d, \phi_c, \phi_l) \qquad (2)$$
$$|A_F| <= C$$
$$P(a \mid w, \phi_s, \phi_d, \phi_c, \phi_l) > T$$

where C and T are two parameters used to threshold the number of candidate argument labels for each word and their probabilities respectively. For each candidate argument we consider the top C argument labels that have probability above T as determined during argument labeling[7].

Arguments for each frame are identified and fitted independently of the other frames. This allows argument sharing across frames, such that the same head word can be an argument of multiple frames. For example, the sentence *"The boy met the girl that he loves"* has two verbal predicates *met* and *loves* that share the same arguments *boy* and *girl*. However, independent assignment of frames and arguments does not take into account the relationships between frames and arguments in a single sentence. In future work, we plan to investigate joint labeling of all targets and their corresponding attributes in a sentence using tree kernels (Croce, Basili, and Moschitti 2015).

Domain Adaptation

We use two methods for domain adaptation: 1) domain configuration and 2) model interpolation.

Domain configuration is achieved through modifying the UT and LA configuration tables for unambiguous targets and frame arguments respectively. Target words that are ambiguous in FrameNet may be unambiguous in a particular domain. For example, in our Communicator data (see next section), the verb *leave* is consistently annotated as a *Departing* frame, while in FrameNet it has six additional frames, including *Causation, Giving,* and *Quitting*. Similarly, frame arguments observed in a domain may be fewer than those in FrameNet. For example, the *Departing* frame occurs with 9 different arguments in Communicator while its FrameNet definition lists 24 different arguments. A dialog system developer can construct UT and LA while designing the dialog system (cf. (Bastianelli et al. 2014)).

For *model interpolation*, a small amount of domain-specific data is added to the FrameNet data prior to model training. We used 10% of the unique Communicator sentences, about 300 sentences, which can be annotated by a trained labeler within two days[8]. We plan to investigate more sophisticated models for domain adaptation in future work.

[6] The LA table is automatically generated from FrameNet XML frame definitions.

Dataset	#Frames unique	#Sent	#Targets total	#Args total
FrameNet LU	866	150K	150K	306K
FrameNet FT	706	4K	20K	42K
Communicator	80	3.2K	3.3K	7.5K

Table 2: Evaluation datasets

Data

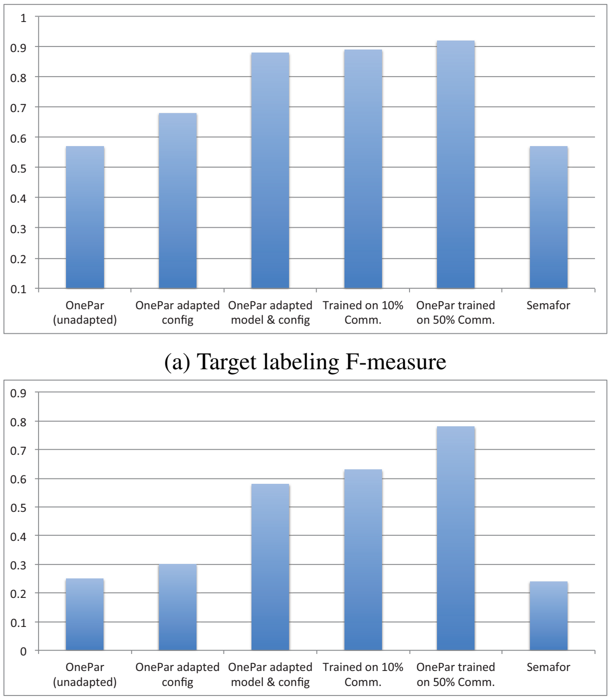
Table 2 summarizes the datasets used in our experiments. The FrameNet dataset (Lowe, Baker, and Fillmore 1997) contains 150K sentences with selective annotations of lexical units (LU) and 4K sentences with full text annotations of all predicates in each sentence (FT). The FrameNet data includes a total of 1K unique frames. The Communicator 2000 corpus consists of 662 human-computer spoken (telephone) dialogs in a travel booking domain. We annotated FrameNet verb frames on the 3.2K unique sentences resulting from automatic sentence splitting of Communicator utterances. The corpus includes 80 unique frames.

Results

Table 3 presents evaluation results comparing generic, domain-adapted and domain-specific models. We report precision, recall, and F-measure for target labeling (Lbl-p, Lbl-r, Lbl-f columns left), argument identification (Id-p, Id-r, Id-f) and argument labeling (Lbl-p, Lbl-r, Lbl-f columns right). For Communicator data all words labeled as *verb* by our POS tagger are identified as targets. For FrameNet data targets are identified and labeled by a joint model. Arguments are labeled on automatically identified targets to estimate the expected performance of a full system. For all experiments, we use a single test split of 1.6K utterances selected at random from the entire Communicator corpus.

We report two types of argument evaluation: *strict* and *lenient*. For strict evaluation, we measure exact match between labeled and identified argument head words. Recall that *OnePar* labels head words in a dependency parse, while FrameNet reference annotations label word spans. Because argument head identification is noisy due to errors in dependency parsing, strict evaluation can erroneously penalize correct predictions. For lenient evaluation, we consider argument ID and labeling to be correct if a labeled argument head word matches any word inside the corresponding argument span in the annotation.

On FrameNet data, the generic model achieves F-measures for target and lenient argument labeling of 0.64 and 0.44 respectively (row 1). However, on Communicator data, F-measures decrease to 0.57/0.25 (row 2), which is comparable to the out-of-the-box performance of SEMAFOR on this data (row 7)[9]. With domain configuration to the Communicator domain, F-measures increase somewhat to 0.68/0.30 (row 3). With domain configuration and domain adaptation using only about 300 sentences, F-measures increase dramatically to 0.88/0.58 (row 4). Figures 3a and 3b highlight the effect of domain adaptation on the F-measure for the target and argument labeling tasks.



(a) Target labeling F-measure

(b) Lenient argument labeling F-measure

Figure 3: *OnePar* system results

For comparison, we trained domain-specific models for Communicator. A domain-specific model trained on 10% of the Communicator data (300 sentences) achieves F-measures for target and argument labeling of 0.89 and 0.63 (row 5), beating the generic model with domain configuration and adaptation. With a larger training set of 50% of the Communicator data (1.5K utterances), performance increases further to 0.92/0.78[10]. Although the performance of domain-specific models beats that of the generic model with domain configuration and adaptation, the domain-specific models can only handle travel booking. If a new domain is added to the dialog system (as in the recent third dialog state tracking challenge, http://camdial.org/∼mh521/dstc/), the system should be able to do some SLU quickly. The generic model with domain configuration and adaptation represents a reasonable compromise.

Discussion

Our experimental results show poor performance of generic models on domain-specific spoken language utterances. This may be one of the reasons why generic semantic parsing,

[7] We use C=2 and T=0.2, values optimized on FrameNet data.

[8] We estimate the Communicator corpus annotation to take five minutes per sentence.

[9] To convert SEMAFOR's argument span prediction to SDL format, argument span head nodes on SEMAFOR's predicted spans are identified using the method described earlier in this paper.

[10] Similar performance for target labeling on Italian dialog data was reported in (Coppola et al. 2008).

	Model	Conf.	Lbl-p	Lbl-r	Lbl-f	Id-p	Id-r	Id-f	Lbl-p	Lbl-r	Lbl-f
OnePar			Target Eval on FrameNet			Lenient Argument Eval on FrameNet					
1	F	F	0.62	0.67	0.64	0.58	0.52	0.55	0.47	0.42	0.44
OnePar			Target Eval on Communicator			Lenient Argument Eval on Communicator					
2	F	F	0.60	0.55	0.57	0.69	0.44	0.53	0.33	0.21	0.25
3	F	C	0.71	0.65	0.68	0.69	0.34	0.46	0.46	0.23	0.30
4	F+C10%	C	0.92	0.85	0.88	0.88	0.58	0.70	0.73	0.48	0.58
5	C 10%	C	0.93	0.85	0.89	0.94	0.67	0.78	0.75	0.54	0.63
6	C 50%	C	0.95	0.88	0.92	0.96	0.85	0.90	0.83	0.73	0.78
7	SEMAFOR		0.62	0.53	0.57	0.70	0.37	0.48	0.36	0.19	0.24
OnePar						Strict Argument Eval on Communicator					
8	F	F				0.54	0.34	0.42	0.27	0.17	0.21
9	F	C				0.56	0.27	0.37	0.38	0.19	0.25
10	F+C10%	C				0.68	0.44	0.54	0.58	0.38	0.46
11	C 10%	C				0.70	0.50	0.58	0.60	0.43	0.50
12	C 50%	C				0.72	0.64	0.68	0.66	0.59	0.62
13	SEMAFOR					0.55	0.28	0.37	0.29	0.15	0.20

Table 3: SDL performance results for OnePar and SEMAFOR. OnePar's models and the configuration files for *UT* and *LA* are trained on FrameNet (F), Communicator (C), or both datasets. Precision/recall for target identification (whether a word is a target) for OnePar on FrameNet data is 0.71/0.76 and on Communicator, 1.0/0.91. Precision/recall for target ID for SEMAFOR on FrameNet is 1.0/0.86.

for which systems have existed for ten years, is not widely adapted for SLU. Rule-based domain configuration with a generic model can improve performance, and the addition of a small amount of domain-specific training data can give results close to those achievable by a domain-specific model.

It is surprising that a model trained on only a small amount of domain specific data (lines 5 and 11 in Table 3) outperforms a hybrid model with the same amount of domain-specific data (lines 4 and 9). The biggest performance gain from the domain specific model comes from increased recall in argument identification. While FrameNet frames are conceptually generic, their instantiation may differ across domains. Less lexicalized feature sets may help.

Conclusions

In this paper, we consider an essential issue for SLU: how to obtain accuracy in-domain with rapid adaptability to new domains, without having to collect and label large amounts of data for each new domain. Our main contributions are: (1) a description of *OnePar*, a novel system for FrameNet-style semantic dependency labeling (SDL), and (2) a detailed comparative assessment of methods for domain adaptation of SDL, conducted using new annotations on the Communicator 2000 corpus, which we will release to the research community.

In this paper, we experiment with two separate but related concepts: *generic representations* (FrameNet) and *generic resources* (data annotated with FrameNet frames). We show that to obtain good in-domain SLU performance, generic resources must be adapted and augmented with some domain-specific data. Nevertheless, using a generic representation like FrameNet provides several benefits. First, it provides a standardized, linguistically motivated representation across domains, opening the possibility for reuse of resources. Second, a generic model trained on generic resources can be used to bootstrap an initial version of an SLU, overcoming "the chicken and egg" problem where data is needed to build a system, and a system is needed to collect data. The amount of additional domain-specific training data is small. Lastly, frame-semantic representations can be used as a human-readable communication protocol to enable interoperability between AI systems running in the cloud. A personal assistant needs to have a wide range of capabilities that constantly evolve as new products and services become available. The back-end functions of native applications (such as hotel reservation or cab booking) can be mapped into generic frame representations. A generic SLU would capture natural language requests for these services. A generic frame-semantic representation would allow subscription-based functionality expansion for such systems much as schema.org allows data sharing for web services.

In future work, we will explore more sophisticated methods of domain adaptation, including model interpolation and domain-similar data selection. We will also evaluate *OnePar* for SLU in a real dialog system.

References

Artzi, Y., and Zettlemoyer, L. 2011. Bootstrapping semantic parsers from conversations. In *Proceedings of EMNLP*.

Baker, C. F.; Fillmore, C. J.; and Lowe, J. B. 1998. The Berkeley FrameNet project. In *Proceedings of COLING-ACL*.

Bastianelli, E.; Castellucci, G.; Croce, D.; Basili, R.; and Nardi, D. 2014. Effective and robust natural language understanding for human-robot interaction. In *Proceedings of ECAI*.

Bauer, D.; Fürstenau, H.; and Rambow, O. C. 2012. The dependency-parsed FrameNet corpus. In *Proceedings of LREC*.

Carreras, X., and Màrquez, L. 2005. Introduction to the CoNLL-2005 shared task: Semantic role labeling. In *Proceedings of CoNLL*.

Chen, Y.-N.; Wang, W. Y.; and Rudnicky, A. I. 2013. Unsupervised induction and filling of semantic slots for spoken dialogue systems using frame-semantic parsing. In *Proceedings of ASRU*.

Coppola, B.; Moschitti, A.; Tonelli, S.; and Riccardi, G. 2008. Automatic FrameNet-based annotation of conversational speech. In *Proceedings of SLT*.

Coppola, B.; Moschitti, A.; and Riccardi, G. 2009. Shallow semantic parsing for spoken language understanding. In *Proceedings of HLT/NAACL*.

Croce, D., and Basili, R. 2011. Structured learning for semantic role labeling. In *Proceedings of AI*IA*.

Croce, D.; Basili, R.; and Moschitti, A. 2015. Semantic tree kernels for statistical natural language learning. In *Harmonization and Development of Resources and Tools for Italian Natural Language Processing within the PARLI Project*. Springer International Publishing. 93–113.

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

Das, D.; Chen, D.; Martins, A. F.; Schneider, N.; and Smith, N. A. 2014. Frame-semantic parsing. *Computational Linguistics* 40(1):9–56.

Dzikovska, M. O.; Allen, J. F.; and Swift, M. D. 2008. Linking semantic and knowledge representations in a multidomain dialogue system. *Journal of Logic and Computation* 18(3):405–430.

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

Hajič, J.; Ciaramita, M.; Johansson, R.; Kawahara, D.; Martí, M. A.; Màrquez, L.; Meyers, A.; Nivre, J.; Padó, S.; Štěpánek, J.; et al. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of CoNLL*.

Huang, F., and Yates, A. 2010. Open-domain semantic role labeling by modeling word spans. In *Proceedings of the ACL*.

Johansson, R., and Nugues, P. 2007. LTH: Semantic structure extraction using nonprojective dependency trees. In *Proceedings of the 4th International Workshop on Semantic Evaluations*.

Kingsbury, P.; Palmer, M.; and Marcus, M. 2002. Adding semantic annotation to the Penn TreeBank. In *Proceedings of HLT*.

Kipper, K.; Korhonen, A.; Ryant, N.; and Palmer, M. 2008. A large-scale classification of English verbs. *Language Resources and Evaluation Journal* 42(1):21–40.

Lang, J., and Lapata, M. 2014. Similarity-driven semantic role induction via graph partitioning. *Computational Linguistics* 40(3):633–669.

Lowe, J. B.; Baker, C. F.; and Fillmore, C. J. 1997. A frame-semantic approach to semantic annotation. In *Proceedings of the ACL - SIGLEX Workshop*.

Marcus, M. P.; Santorini, B.; and Marcinkiewicz, M. A. 1993. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics* 19(2):313–330.

Punyakanok, V.; Roth, D.; and Yih, W. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics* 34(2):257–287.

Surdeanu, M.; Johansson, R.; Meyers, A.; Mrquez, L.; and Nivre, J. 2008. The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of CoNLL*.

Täckström, O.; Ganchev, K.; and Das, D. 2015. Efficient inference and structured learning for semantic role labeling. *Transactions of the Association for Computational Linguistics* 3:29–41.

Tur, G.; Hakkani-Tür, D.; and Chotimongkol, A. 2005. Semi-supervised learning for spoken language understanding semantic role labeling. In *Proceedings of ASRU*. IEEE.

Walker, M., et al. 2001. DARPA Communicator dialog travel planning systems: The June 2000 data collection. In *Proceedings of Eurospeech*.

Williams, J.; Kamal, E.; Ashour, M.; Amr, H.; Miller, J.; and Zweig, G. 2015. Fast and easy language understanding for dialog systems with microsoft language understanding intelligent service (LUIS). In *Proceedings of SIGDIAL*.

Wit.AI. 2015. Natural language for developers. https://wit.ai.