Leveraging Domain-Independent Information in Semantic Parsing

Dan Goldwasser
University of Maryland
College Park, MD 20740
goldwas1@umiacs.umd.edu

Dan Roth
University of Illinois
Urbana, IL 61801
danr@illinois.edu

Abstract

Semantic parsing is a domain-dependent process by nature, as its output is defined over a set of domain symbols. Motivated by the observation that interpretation can be decomposed into domain-dependent and independent components, we suggest a novel interpretation model, which augments a domain dependent model with abstract information that can be shared by multiple domains. Our experiments show that this type of information is useful and can reduce the annotation effort significantly when moving between domains.

1 Introduction

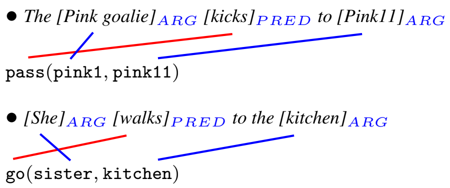
Natural Language (NL) understanding can be intuitively understood as a general capacity, mapping words to entities and their relationships. However, current work on automated NL understanding (typically referenced as *semantic parsing* (Zettlemoyer and Collins, 2005; Wong and Mooney, 2007; Chen and Mooney, 2008; Kwiatkowski et al., 2010; Börschinger et al., 2011)) is restricted to a given output domain[1] (or task) consisting of a closed set of meaning representation symbols, describing domains such as robotic soccer, database queries and flight ordering systems.

In this work, we take a first step towards constructing a semantic interpreter that can leverage information from multiple tasks. This is not a straight forward objective – the domain specific nature of semantic interpretation, as described in the current literature, does not allow for an easy move between domains. For example, a system trained for the task of understanding database queries will not be of any use when it will be given a sentence describing robotic soccer instructions.

In order to understand this difficulty, a closer look at semantic parsing is required. Given a sentence, the interpretation process breaks it into a set of interdependent decisions, which rely on an underlying representation mapping words to symbols and syntactic patterns into compositional decisions. This representation takes into account domain specific information (e.g., a lexicon mapping phrases to a domain predicate) and is therefore of little use when moving to a different domain.

In this work, we attempt to develop a domain independent approach to semantic parsing. We do it by developing a layer of representation that is applicable to multiple domains. Specifically, we add an intermediate layer capturing shallow semantic relations between the input sentence constituents. Unlike semantic parsing which maps the input to a closed set of symbols, this layer can be used to identify general predicate-argument structures in the input sentence.The following example demonstrates the key idea behind our representation – two sentences from two different domains have a similar intermediate structure.

Example 1. *Domains with similar intermediate structures*



- *The [Pink goalie]*$_{ARG}$ *[kicks]*$_{PRED}$ *to [Pink11]*$_{ARG}$

pass(pink1, pink11)

- *[She]*$_{ARG}$ *[walks]*$_{PRED}$ *to the [kitchen]*$_{ARG}$

go(sister, kitchen)

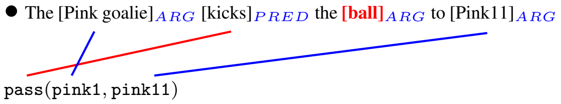
In this case, the constituents of the first sentence (from the Robocup domain (Chen and Mooney, 2008)), are assigned domain-independent predicate-argument labels (e.g., the word corresponding to a logical function is identified as a $PRED$). Note that it does not use any domain specific information, for example, the $PRED$ label assigned to the word *"kicks"* indicates that this word is the predicate of the sentence, not a specific domain predicate (e.g., pass$(\cdot)$). The intermediate layer can be reused across domains. The logical output associated with the second sentence is taken from a different domain, using a different set of output symbols, however it shares the same predicate-argument structure.

Despite the idealized example, in practice,

[1] The term *domain* is overloaded in NLP; in this work we use it to refer to the set of output symbols.

Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 462–466,
Sofia, Bulgaria, August 4-9 2013. ©2013 Association for Computational Linguistics

leveraging this information is challenging, as the logical structure is assumed to only weakly correspond to the domain-independent structure, a correspondence which may change in different domains. The mismatch between the domain independent (linguistic) structure and logical structures typically stems from technical considerations, as the domain logical language is designed according to an application-specific logic and not according to linguistic considerations. This situation is depicted in the following example, in which one of the domain-independent labels is omitted.


● The [Pink goalie]$_{ARG}$ [kicks]$_{PRED}$ the [**ball**]$_{ARG}$ to [Pink11]$_{ARG}$
pass(pink1, pink11)

In order to overcome this difficulty, we suggest a flexible model that is able to leverage the supervision provided in one domain to learn an abstract intermediate layer, and show empirically that it learns a robust model, improving results significantly in a second domain.

2 Semantic Interpretation Model

Our model consists of both domain-dependent (mapping between text and a closed set of symbols) and domain independent (abstract predicate-argument structures) information. We formulate the joint interpretation process as a structured prediction problem, mapping a NL input sentence ($\mathbf{x}$), to its highest ranking interpretation and abstract structure ($\mathbf{y}$). The decision is quantified using a linear objective, which uses a vector w, mapping features to weights and a feature function Φ which maps the output decision to a feature vector. The output interpretation $\mathbf{y}$ is described using a subset of first order logic, consisting of typed constants (e.g., robotic soccer player), functions capturing relations between entities, and their properties (e.g., pass(x, y), where pass is a function symbol and x, y are typed arguments). We use data taken from two grounded domains, describing robotic soccer events and household situations.

We begin by formulating the domain-specific process. We follow (Goldwasser et al., 2011; Clarke et al., 2010) and formalize semantic inference as an Integer Linear Program (ILP). Due to space consideration, we provide a brief description (see (Clarke et al., 2010) for more details). We then proceed to augment this model with domain-independent information, and connect the two models by constraining the ILP model.

2.1 Domain-Dependent Model

Interpretation is composed of several decisions, capturing mapping of input tokens to logical fragments (first order) and their composition into larger fragments (second). We encode a first-order decision as α_{cs}, a binary variable indicating that constituent c is aligned with the logical symbol s. A second-order decision $\beta_{cs,dt}$, is encoded as a binary variable indicating that the symbol t (associated with constituent d) is an argument of a function s (associated with constituent c). The overall inference problem (Eq. 1) is as follows:

$$F_{\mathbf{w}}(\mathbf{x}) = \arg\max_{\alpha,\beta} \sum_{c\in\mathbf{x}} \sum_{s\in D} \alpha_{cs} \cdot \mathbf{w}^T \Phi_1(\mathbf{x}, c, s) + \sum_{c,d\in\mathbf{x}} \sum_{s,t\in D} \beta_{cs,dt} \cdot \mathbf{w}^T \Phi_2(\mathbf{x}, c, s, d, t) \quad (\mathbf{1})$$

We restrict the possible assignments to the decision variables, forcing the resulting output formula to be *syntactically* legal, for example by restricting active β-variables to be type consistent, and forcing the resulting functional composition to be acyclic and fully connected (we refer the reader to (Clarke et al., 2010) for more details). We take advantage of the flexible ILP framework and encode these restrictions as global constraints.

Features We use two types of feature, first-order Φ_1 and second-order Φ_2. Φ_1 depends on lexical information: each mapping of a lexical item c to a domain symbol s generates a feature. In addition each combination of a lexical item c and an symbol type generates a feature.

Φ_2 captures a pair of symbols and their alignment to lexical items. Given a second-order decision $\beta_{cs,dt}$, a feature is generated considering the normalized distance between the head words in the constituents c and d. Another feature is generated for every composition of symbols (ignoring the alignment to the text).

2.2 Domain-Independent Information

We enhance the decision process with information that abstracts over the attributes of specific domains by adding an intermediate layer consisting of the predicate-argument structure of the sentence. Consider the mappings described in Example 1. Instead of relying on the mapping between *Pink goalie* and pink1, this model tries to identify an ARG using different means. For example, the fact that it is preceded by a determiner, or capitalized provide useful cues. We do not assume any language specific knowledge and use features that help capture these cues.

This information is used to assist the overall learning process. We assume that these labels correspond to a binding to some logical symbol, and encode it as a constraint forcing the relations between the two models. Moreover, since learning this layer is a by-product of the learning process (as it does not use any labeled data) forcing the connection between the decisions is the mechanism that drives learning this model.

Our domain-independent layer bears some similarity to other semantic tasks, most notably Semantic-Role Labeling (SRL) introduced in (Gildea and Jurafsky, 2002), in which identifying the predicate-argument structure is considered a preprocessing step, prior to assigning argument labels. Unlike SRL, which aims to identify linguistic structures alone, in our framework these structures capture both natural-language and domain-language considerations.

Domain-Independent Decision Variables We add two new types of decisions abstracting over the domain-specific decisions. We encode the new decisions as γ_c and δ_{cd}. The first (γ) captures local information helping to determine if a given constituent c is likely to have a label (i.e., γ_c^P for predicate or γ_c^A for argument). The second (δ) considers higher level structures, quantifying decisions over both the labels of the constituents `c,d` as a predicate-argument pair. Note, a given word c can be labeled as PRED or ARG if γ_c and δ_{cd} are active.

Model's Features We use the following features: (1) **Local Decisions** $\Phi_3(\gamma(c))$ use a feature indicating if c is capitalized, a set of features capturing the context of c (window of size 2), such as determiner and quantifier occurrences. Finally we use a set of features capturing the suffix letters of c, these features are useful in identifying verb patterns. Features indicate if c is mapped to an ARG or PRED. (2) **Global Decision** $\Phi_4(\delta(c,d))$: a feature indicating the relative location of c compared to d in the input sentence. Additional features indicate properties of the relative location, such as if the word appears initially or finally in the sentence.

Combined Model In order to consider both types of information we augment our decision model with the new variables, resulting in the following objective function (Eq. 2).

$$F_{\mathbf{w}}(\mathbf{x}) = \arg\max_{\alpha,\beta} \sum_{c\in\mathbf{x}} \sum_{s\in D} \alpha_{cs} \cdot \mathbf{w_1}^T \Phi_1(\mathbf{x}, c, s) + \sum_{c,d\in\mathbf{x}} \sum_{s,t\in D} \sum_{i,j} \beta_{cs^i,dt^j} \cdot \mathbf{w_2}^T \Phi_2(\mathbf{x}, c, s^i, d, t^j) + \sum_{c\in\mathbf{x}} \gamma_c \cdot \mathbf{w_3}^T \Phi_3(\mathbf{x}, c) + \sum_{c,d\in\mathbf{x}} \delta_{cd} \cdot \mathbf{w_4}^T \Phi_4(\mathbf{x}, c, d) \quad \textbf{(2)}$$

For notational convenience we decompose the weight vector $\mathbf{w}$ into four parts, $\mathbf{w_1}, \mathbf{w_2}$ for features of (first, second) order domain-dependent decisions, and similarly for the independent ones. In addition, we also add new constraints tying these new variables to semantic interpretation :

$\forall c \in x \; (\gamma_c \rightarrow \alpha_{c,s^1} \vee \alpha_{c,s^2} \vee ... \vee \alpha_{c,s^n})$

$\forall c \in x, \; \forall d \in x \; (\delta_{c,d} \rightarrow \beta_{c,s^1,dt^1} \vee \beta_{c,s^2,dt^1} \vee ... \vee \beta_{c,s^n,dt^n})$

(where n is the length of x).

2.3 Learning the Combined Model

The supervision to the learning process is given via data consisting of pairs of sentences and (domain specific) semantic interpretation. Given that we have introduced additional variables that capture the more abstract predicate-argument structure of the text, we need to induce these as latent variables. Our decision model maps an input sentence x, into a logical output y and predicate-argument structure h. We are only supplied with training data pertaining to the input (x) and output (y). We use a variant of the latent structure perceptron to learn in these settings[2].

3 Experimental Settings

Situated Language This dataset, introduced in (Bordes et al., 2010), describes situations in a simulated world. The dataset consists of triplets of the form - (x,u, y), where x is a NL sentence describing a situation (e.g., *"He goes to the kitchen"*), u is a world state consisting of grounded relations (e.g., `loc(John, Kitchen)`) description, and y is a logical interpretation corresponding to x.

The original dataset was used for *concept tagging*, which does not include a compositional aspect. We automatically generated the full logical structure by mapping the constants to function arguments. We generated additional function symbols of the same relation, but of different arity when needed [3]. Our new dataset consists of 25 relation symbols (originally 15). In our experiments we used a set of 5000 of the training triplets.

Robocup The Robocup dataset, originally introduced in (Chen and Mooney, 2008), describes robotic soccer events. The dataset was collected for the purpose of constructing semantic parsers from ambiguous supervision and consists of both "noisy" and gold labeled data. The noisy dataset

[2] Details omitted, see (Chang et al., 2010) for more details.

[3] For example, a unary relation symbol for *"He plays"*, and a binary for *"He plays with a ball"*.

System	Training Procedure
Dom-Init	$\mathbf{w_1}$: Noisy probabilistic model, described below.
Pred-Arg$_S$	**Only** $\mathbf{w_3}, \mathbf{w_4}$ Trained over the Situ. dataset.
Combined$_{RL}$	$\mathbf{w_1}, \mathbf{w_2}, \mathbf{w_3}, \mathbf{w_4}$:learned from Robocup gold
Combined$_{RI+S}$	$\mathbf{w_3}, \mathbf{w_4}$: learned from the Situ. dataset, $\mathbf{w_1}$ uses the Dom-Init Robocup model.
Combined$_{RL+S}$	$\mathbf{w_3}, \mathbf{w_4}$: Initially learned over the Situ. dataset, updated jointly with $\mathbf{w_1}, \mathbf{w_2}$ over Robocup gold

Table 1: Evaluated System descriptions.

System	Matching	Parsing
Pred-Arg$_S$	0.692	–
Dom-Init	0.823	0.357
Combined$_{RI+S}$	**0.905**	**0.627**
(Börschinger et al., 2011)	–	0.86
(Kim and Mooney, 2010)	0.885	0.742

Table 2: Results for the matching and parsing tasks. Our system performs well on the matching task **without any domain information**. Results for both parsing and matching tasks show that using domain-independent information improves results dramatically.

was constructed by temporally aligning a stream of soccer events occurring during a robotic soccer match with human commentary describing the game. This dataset consists of pairs $(\mathtt{x}, \{\mathtt{y_0}, \mathtt{y_k}\})$, $\mathtt{x}$ is a sentence and $\{\mathtt{y_0}, \mathtt{y_k}\}$ is a set of events (logical formulas). One of these events is assumed to correspond to the comment, however this is not guaranteed. The gold labeled labeled data consists of pairs $(\mathtt{x}, \mathtt{y})$. The data was collected from four Robocup games. In our experiments we follow other works and use 4-fold cross validation, training over 3 games and testing over the remaining game. We evaluate the Accuracy of the parser over the test game data.[4] Due to space considerations, we refer the reader to (Chen and Mooney, 2008) for further details about this dataset.

Semantic Interpretation Tasks We consider two of the tasks described in (Chen and Mooney, 2008) (1) *Semantic Parsing* requires generating the correct logical form given an input sentence. (2) *Matching*, given a NL sentence and a set of several possible interpretation candidates, the system is required to identify the correct one. In all systems, the source for domain-independent information is the Situated domain, and the results are evaluated over the Robocup domain.

Experimental Systems We tested several variations, all solving Eq. 2, however different resources were used to obtain Eq. 2 parameters (see sec. 2.2). Tab. 1 describes the different variations. We used the noisy Robocup dataset to initialize **Dom-Init**, a noisy probabilistic model, constructed by taking statistics over the noisy robocup data and computing $\mathtt{p}(\mathtt{y}|\mathtt{x})$. Given the training set $\{(\mathtt{x}, \{\mathtt{y_1}, .., \mathtt{y_k}\})\}$, every word in $\mathtt{x}$ is aligned to every symbol in every $\mathtt{y}$ that is aligned with it. The probability of a matching $(\mathtt{x}, \mathtt{y})$is computed as the product: $\prod_{i=1}^{n} \mathtt{p}(\mathtt{y_i}|\mathtt{x_i})$, where n is the number of symbols appearing in y, and x_i, y_i is the word level matching to a logical symbol. *Note that this model uses lexical information only.*

4 Knowledge Transfer Experiments

We begin by studying the role of domain-independent information when very little domain information is available. Domain-independent information is learned from the situated domain and domain-specific information (Robocup) available is the simple probabilistic model (Dom-Init). This model can be considered as a noisy probabilistic lexicon, without any domain-specific compositional information, *which is only available through domain-independent information.*

The results, summarized in Table 2, show that in both tasks domain-independent information is extremely useful and can make up for missing domain information. Most notably, performance for the matching task using only domain independent information (Pred-Arg$_S$) was surprisingly good, with an accuracy of 0.69. Adding domain-specific lexical information (Combined$_{RI+S}$) pushes this result to over 0.9, currently the highest for this task – achieved without domain specific learning.

The second set of experiments study whether using domain independent information, when relevant (gold) domain-specific training data is available, improves learning. In this scenario, the domain-independent model is updated according to training data available for the Robocup domain. We compare two system over varying amounts of training data (25, 50, 200 training samples and the full set of 3 Robocup games), one bootstrapped using the Situ. domain (Combined$_{RL+S}$) and one relying on the Robocup training data alone (Combined$_{RL}$). The results, summarized in table 3, consistently show that transferring domain independent information is helpful, and helps push the learned models beyond the supervision offered by the relevant domain training data. Our final system, trained over the entire dataset achieves a

[4] In our model accuracy is equivalent to F-measure.

System	# training	Parsing
Combined$_{RL+S}$(Combined$_{RL}$)	25	0.16 (0.03)
Combined$_{RL+S}$(Combined$_{RL}$)	50	0.323 (0.16)
Combined$_{RL+S}$(Combined$_{RL}$)	200	0.385 (0.36)
Combined$_{RL+S}$(Combined$_{RL}$)	full game	**0.86** (0.79)
(Chen et al., 2010)	full game	0.81

Table 3: Evaluating our model in a learning settings. The domain-independent information is used to bootstrap learning from the Robocup domain. Results show that this information improves performance significantly, especially when little data is available

score of 0.86, significantly outperforming (Chen et al., 2010), a competing supervised model. It achieves similar results to (Börschinger et al., 2011), the current state-of-the-art for the parsing task over this dataset. The system used in (Börschinger et al., 2011) learns from ambiguous training data and achieves this score by using global information. We hypothesize that it can be used by our model and leave it for future work.

5 Conclusions

In this paper, we took a first step towards a new kind of generalization in semantic parsing: constructing a model that is able to generalize to a new domain defined over a different set of symbols. Our approach adds an additional hidden layer to the semantic interpretation process, capturing shallow but domain-independent semantic information, which can be shared by different domains. Our experiments consistently show that domain-independent knowledge can be transferred between domains. We describe two settings; in the first, where only noisy lexical-level domain-specific information is available, we observe that the model learned in the other domain can be used to make up for the missing compositional information. For example, in the matching task, even when no domain information is available, identifying the abstract predicate argument structure provides sufficient discriminatory power to identify the correct event in over 69% of the times.

In the second setting domain-specific examples are available. The learning process can still utilize the transferred knowledge, as it provides scaffolding for the latent learning process, resulting in a significant improvement in performance.

6 Acknowledgement

The authors would like to thank Julia Hockenmaier, Gerald DeJong, Raymond Mooney and the anonymous reviewers for their efforts and insightful comments.

Most of this work was done while the first author was at the University of Illinois. The authors gratefully acknowledge the support of the Defense Advanced Research Projects Agency (DARPA) Machine Reading Program under Air Force Research Laboratory (AFRL) prime contract no. FA8750-09-C-0181. In addition, this material is based on research sponsored by DARPA under agreement number FA8750-13-2-0008. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA,AFRL, or the U.S. Government.

References

A. Bordes, N. Usunier, R. Collobert, and J. Weston. 2010. Towards understanding situated natural language. In *AISTATS*.

B. Börschinger, B. K. Jones, and M. Johnson. 2011. Reducing grounded learning tasks to grammatical inference. In *EMNLP*.

M. Chang, D. Goldwasser, D. Roth, and V. Srikumar. 2010. Discriminative learning over constrained latent representations. In *NAACL*.

D. Chen and R. Mooney. 2008. Learning to sportscast: a test of grounded language acquisition. In *ICML*.

D. L. Chen, J. Kim, and R. J. Mooney. 2010. Training a multilingual sportscaster: Using perceptual context to learn language. *Journal of Artificial Intelligence Research*, 37:397–435.

J. Clarke, D. Goldwasser, M. Chang, and D. Roth. 2010. Driving semantic parsing from the world's response. In *CoNLL*.

D. Gildea and D. Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*.

D. Goldwasser, R. Reichart, J. Clarke, and D. Roth. 2011. Confidence driven unsupervised semantic parsing. In *ACL*.

J. Kim and R. J. Mooney. 2010. Generative alignment and semantic parsing for learning from ambiguous supervision. In *COLING*.

T. Kwiatkowski, L. Zettlemoyer, S. Goldwater, , and M. Steedman. 2010. Inducing probabilistic ccg grammars from logical form with higher-order unification. In *EMNLP*.

Y.W. Wong and R. Mooney. 2007. Learning synchronous grammars for semantic parsing with lambda calculus. In *ACL*.

L. Zettlemoyer and M. Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammars. In *UAI*.