A Bayesian Model for Unsupervised Semantic Parsing

Ivan Titov
Saarland University
Saarbruecken, Germany
titov@mmci.uni-saarland.de

Alexandre Klementiev
Johns Hopkins University
Baltimore, MD, USA
aklement@jhu.edu

Abstract

We propose a non-parametric Bayesian model for unsupervised semantic parsing. Following Poon and Domingos (2009), we consider a semantic parsing setting where the goal is to (1) decompose the syntactic dependency tree of a sentence into fragments, (2) assign each of these fragments to a cluster of semantically equivalent syntactic structures, and (3) predict predicate-argument relations between the fragments. We use hierarchical Pitman-Yor processes to model statistical dependencies between meaning representations of predicates and those of their arguments, as well as the clusters of their syntactic realizations. We develop a modification of the Metropolis-Hastings split-merge sampler, resulting in an efficient inference algorithm for the model. The method is experimentally evaluated by using the induced semantic representation for the question answering task in the biomedical domain.

1 Introduction

Statistical approaches to semantic parsing have recently received considerable attention. While some methods focus on predicting a complete formal representation of meaning (Zettlemoyer and Collins, 2005; Ge and Mooney, 2005; Mooney, 2007), others consider more shallow forms of representation (Carreras and Màrquez, 2005; Liang et al., 2009). However, most of this research has concentrated on *supervised* methods requiring large amounts of labeled data. Such annotated resources are scarce, expensive to create and even the largest of them tend to have low coverage (Palmer and Sporleder, 2010), motivating the need for unsupervised or semi-supervised techniques.

Conversely, research in the closely related task of relation extraction has focused on unsupervised or minimally supervised methods (see, for example, (Lin and Pantel, 2001; Yates and Etzioni, 2009)). These approaches cluster semantically equivalent verbalizations of relations, often relying on syntactic fragments as features for relation extraction and clustering (Lin and Pantel, 2001; Banko et al., 2007). The success of these methods suggests that semantic parsing can also be tackled as clustering of syntactic realizations of predicate-argument relations. While a similar direction has been previously explored in (Swier and Stevenson, 2004; Abend et al., 2009; Lang and Lapata, 2010), the recent work of (Poon and Domingos, 2009) takes it one step further by not only predicting predicate-argument structure of a sentence but also assigning sentence fragments to clusters of semantically similar expressions. For example, for a pair of sentences on Figure 1, in addition to inducing predicate-argument structure, they aim to assign expressions *"Steelers"* and *"the Pittsburgh team"* to the same semantic class `Steelers`, and group expressions *"defeated"* and *"secured the victory over"*. Such semantic representation can be useful for entailment or question answering tasks, as an entailment model can abstract away from specifics of syntactic and lexical realization relying instead on the induced semantic representation. For example, the two sentences in Figure 1 have identical semantic representation, and therefore can be hypothesized to be equivalent.

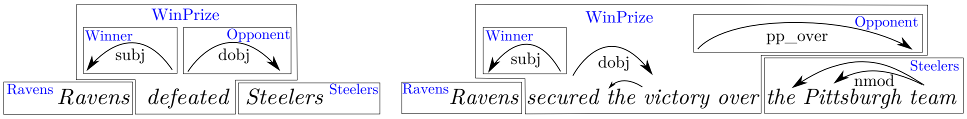

Figure 1: An example of two different syntactic trees with a common semantic representation `WinPrize(Ravens, Steelers)`.

From the statistical modeling point of view, joint learning of predicate-argument structure and discovery of semantic clusters of expressions can also be beneficial, because it results in a more compact model of selectional preference, less prone to the data-sparsity problem (Zapirain et al., 2010). In this respect our model is similar to recent LDA-based models of selectional preference (Ritter et al., 2010; Séaghdha, 2010), and can even be regarded as their recursive and non-parametric extension.

In this paper, we adopt the above definition of unsupervised semantic parsing and propose a Bayesian non-parametric approach which uses hierarchical Pitman-Yor (PY) processes (Pitman, 2002) to model statistical dependencies between predicate and argument clusters, as well as distributions over syntactic and lexical realizations of each cluster. Our non-parametric model automatically discovers granularity of clustering appropriate for the dataset, unlike the parametric method of (Poon and Domingos, 2009) which have to perform model selection and use heuristics to penalize more complex models of semantics. Additional benefits generally expected from Bayesian modeling include the ability to encode prior linguistic knowledge in the form of hyperpriors and the potential for more reliable modeling of smaller datasets. More detailed discussion of relation between the Markov Logic Network (MLN) approach of (Poon and Domingos, 2009) and our non-parametric method is presented in Section 3.

Hierarchical Pitman-Yor processes (or their special case, hierarchical Dirichlet processes) have previously been used in NLP, for example, in the context of syntactic parsing (Liang et al., 2007; Johnson et al., 2007). However, in all these cases the effective size of the state space (i.e., the number of sub-symbols in the infinite PCFG (Liang et al., 2007), or the number of adapted productions in the adaptor grammar (Johnson et al., 2007)) was not very large. In our case, the state space size equals the total number of distinct semantic clusters, and, thus, is expected to be exceedingly large even for moderate datasets: for example, the MLN model induces 18,543 distinct clusters from 18,471 sentences of the GENIA corpus (Poon and Domingos, 2009). This suggests that standard inference methods for hierarchical PY processes, such as Gibbs sampling, Metropolis-Hastings (MH) sampling with uniform proposals, or the structured mean-field algorithm, are unlikely to result in efficient inference: for example in standard Gibbs sampling all thousands of alternatives should be considered at each sampling move. Instead, we use a split-merge MH sampling algorithm, which is a standard and efficient inference tool for non-hierarchical PY processes (Jain and Neal, 2000; Dahl, 2003) but has not previously been used in hierarchical setting. We extend the sampler to include composition-decomposition of syntactic fragments in order to cluster fragments of variables size, as in the example Figure 1, and also include the argument role-syntax alignment move which attempts to improve mapping between semantic roles and syntactic paths for some fixed predicate.

Evaluating unsupervised models is a challenging task. We evaluate our model both qualitatively, examining the revealed clustering of syntactic structures, and quantitatively, on a question answering task. In both cases, we follow (Poon and Domingos, 2009) in using the corpus of biomedical abstracts. Our model achieves favorable results significantly outperforming the baselines, including state-of-the-art methods for relation extraction, and achieves scores comparable to those of the MLN model.

The rest of the paper is structured as follows. Section 2 begins with a definition of the semantic parsing task. Sections 3 and 4 give background on the MLN model and the Pitman-Yor processes, respectively. In Sections 5 and 6, we describe our model and the inference method. Section 7 provides both qualitative and quantitative evaluation. Finally, additional related work is presented in Section 8.

2 Semantic Parsing

In this section, we briefly define the unsupervised semantic parsing task and underlying aspects and assumptions relevant to our model.

Unlike (Poon and Domingos, 2009), we do not use the lambda calculus formalism to define our task but rather treat it as an instance of frame-semantic parsing, or a specific type of semantic role labeling (Gildea and Jurafsky, 2002). The reason for this is two-fold: first, the frame semantics view is more standard in computational linguistics, sufficient to describe induced semantic representation and convenient to relate our method to the previous work. Second, lambda calculus is a considerably more powerful formalism than the predicate-argument structure used in frame semantics, normally supporting quantification and logical connectors (for example, negation and disjunction), neither of which is modeled by our model or in (Poon and Domingos, 2009).

In frame semantics, the meaning of a predicate is conveyed by a *frame*, a structure of related concepts that describes a situation, its participants and properties (Fillmore et al., 2003). Each frame is characterized by a set of semantic roles (frame elements) corresponding to the arguments of the predicate. It is evoked by a frame evoking element (a predicate). The same frame can be evoked by different but semantically similar predicates: for example, both verbs *"buy"* and *"purchase"* evoke frame `Commerce_buy` in FrameNet (Fillmore et al., 2003).

The aim of the semantic role labeling task is to identify all of the frames evoked in a sentence and label their semantic role fillers. We extend this task and treat semantic parsing as recursive prediction of predicate-argument structure and clustering of argument fillers. Thus, parsing a sentence into this representation involves (1) decomposing the sentence into lexical items (one or more words), (2) assigning a cluster label (a semantic frame or a cluster of argument fillers) to every lexical item, and (3) predicting argument-predicate relations between the lexical items. This process is illustrated in Figure 1. For the leftmost example, the sentence is decomposed into three lexical items: *"Ravens"*, *"defeated"* and *"Steelers"*, and they are assigned to clusters `Ravens`, `WinPrize` and `Steelers`, respectively. Then `Ravens` and `Steelers` are selected as a `Winner` and an `Opponent` in the `WinPrize` frame. In this work, we define a joint model for the labeling and argument identification stages. Similarly to core semantic roles in FrameNet, semantic roles are treated as frame-specific in our model, as our model does not try to discover any correspondences between roles in different frames.

As you can see from the above description, frames (which groups predicates with similar meaning such as the `WinPrize` frame in our example) and clusters of argument fillers (`Ravens` and `Steelers`) are treated in our definition in a similar way. For convenience, we will refer to both types of clusters as *semantic classes*.[1]

This definition of semantic parsing is closely related to a realistic relation extraction setting, as both clustering of syntactic forms of relations (or extraction patterns) and clustering of argument fillers for these relations is crucial for automatic construction of knowledge bases (Yates and Etzioni, 2009).

In this paper, we make three assumptions. First, we assume that each lexical item corresponds to a subtree of the syntactic dependency graph of the sentence. This assumption is similar to the adjacency assumption in (Zettlemoyer and Collins, 2005), though ours may be more appropriate for languages with free or semi-free word order, where syntactic structures are inherently non-projective. Second, we assume that the semantic arguments are local in the dependency tree; that is, one lexical item can be a semantic argument of another one only if they are connected by an arc in the dependency tree. This is a slight simplification of the semantic role labeling problem but one often made. Thus, the argument identification and labeling stages consist of labeling each syntactic arc with a semantic role label. In comparison, the MLN model does not explicitly assume contiguity of lexical items and does not make this directionality assumption but their clustering algorithm uses initialization and clusterization moves such that the resulting model also obeys both of these constraints. Third, as in (Poon and Domingos, 2009), we do not model polysemy as we assume

[1] Semantic classes correspond to lambda-form clusters in (Poon and Domingos, 2009) terminology.

that each syntactic fragment corresponds to a single semantic class. This is not a model assumption and is only used at inference as it reduces mixing time of the Markov chain. It is not likely to be restrictive for the biomedical domain studied in our experiments.

As in some of the recent work on learning semantic representations (Eisenstein et al., 2009; Poon and Domingos, 2009), we assume that dependency structures are provided for every sentence. This assumption allows us to construct models of semantics not Markovian within a sequence of words (see for an example a model described in (Liang et al., 2009)), but rather Markovian within a dependency tree. Though we include generation of the syntactic structure in our model, we would not expect that this syntactic component would result in an accurate syntactic model, even if trained in a supervised way, as the chosen independence assumptions are oversimplistic. In this way, we can use a simple generative story and build on top of the recent success in syntactic parsing.

3 Relation to the MLN Approach

The work of (Poon and Domingos, 2009) models joint probability of the dependency tree and its latent semantic representation using Markov Logic Networks (MLNs) (Richardson and Domingos, 2006), selecting parameters (weights of first-order clauses) to maximize the probability of the observed dependency structures. For each sentence, the MLN induces a Markov network, an undirected graphical model with nodes corresponding to ground atoms and cliques corresponding to ground clauses.

The MLN is a powerful formalism and allows for modeling complex interaction between features of the input (syntactic trees) and latent output (semantic representation), however, unsupervised learning of semantics with general MLNs can be prohibitively expensive. The reason for this is that MLNs are undirected models and when learned to maximize likelihood of syntactically annotated sentences, they would require marginalization over semantic representation but also over the entire space of syntactic structures and lexical units. Given the complexity of the semantic parsing task and the need to tackle large datasets, even approximate methods are likely to be infeasible. In order to overcome this problem, (Poon and Domingos, 2009) group parameters and impose local normalization constraints within each group. Given these normalization constraints, and additional structural constraints satisfied by the model, namely that the clauses should be engineered in such a way that they induce tree-structured graphs for every sentence, the parameters can be estimated by a variant of the EM algorithm.

The class of such restricted MLNs is equivalent to the class of directed graphical models over the same set of random variables corresponding to fragments of syntactic and semantic structure. Given that the above constraints do not directly fit into the MLN methodology, we believe that it is more natural to regard their model as a directed model with an underlying generative story specifying how the semantic structure is generated and how the syntactic parse is drawn for this semantic structure. This view would facilitate understanding what kind of features can easily be integrated into the model, simplify application of non-parametric Bayesian techniques and expedite the use of inference techniques designed specifically for directed models. Our approach makes one step in this direction by proposing a non-parametric version of such generative model.

4 Hierarchical Pitman-Yor Processes

The central component of our non-parametric Bayesian model are Pitman-Yor (PY) processes, which are a generalization of the Dirichlet processes (DPs) (Ferguson, 1973). We use PY processes to model distributions of semantic classes appearing as an argument of other semantic classes. We also use them to model distributions of syntactic realizations for each semantic class and distributions of syntactic dependency arcs for argument types. In this section we present relevant background on PY processes. For a more detailed consideration we refer the reader to (Teh et al., 2006).

The Pitman-Yor process over a set S, denoted $PY(\alpha, \beta, H)$, is a stochastic process whose samples G_0 constitute probability measures on partitions of S. In practice, we do not need to draw measures, as they can be analytically marginalized out. The conditional distribution of x_{j+1} given the previous j draws, with G_0 marginalized out, follows (Blackwell and MacQueen, 1973)

$$x_{j+1}|x_1,\dots x_j \sim \sum_{k=1}^{K} \frac{j_k - \beta}{j+\alpha}\delta_{\phi_k} + \frac{K\beta+\alpha}{j+\alpha}H. \quad (1)$$

where $\phi_1,\dots,\phi_K$ are K values assigned to $x_1, x_2, \dots, x_j$. The number of times ϕ_k was assigned is denoted j_k, so that $j = \sum_{k=1}^{K} j_k$. The parameter $\beta < 1$ controls how heavy the tail of the distribution is: when it approaches 1, a new value is assigned to every draw, when $\beta = 0$ the PY process reduces to DP. The expected value of K scales as $O(\alpha n^{\beta})$ with the number of draws n, while it scales only logarithmically for DP processes. PY processes are expected to be more appropriate for many NLP problems, as they model power-law type distributions common for natural language (Teh, 2006).

Hierarchical Dirichlet Processes (HDP) or hierarchical PY processes are used if the goal is to draw several related probability measures for the same set S. For example, they can be used to generate transition distributions of a Markov model, HDP-HMM (Teh et al., 2006; Beal et al., 2002). For such a HMM, the top-level state proportions are drawn from the top-level stick breaking construction $\gamma \sim GEM(\alpha, \beta)$, and then the individual transition distributions for every state $z = 1, 2, \dots$ ϕ_z are drawn from $PY(\gamma, \alpha', \beta')$. The parameters α' and β' control how similar the individual transition distributions ϕ_z are to the top-level state proportions γ, or, equivalently, how similar the transition distributions are to each other.

5 A Model for Semantic Parsing

Our model of semantics associates with each semantic class a set of distributions which govern the generation of corresponding syntactic realizations[2] and the selection of semantic classes for its arguments. Each sentence is generated starting from the root of its dependency tree, recursively drawing a semantic class, its syntactic realization, arguments and semantic classes for the arguments. Below we describe the model by first defining the set of the model parameters and then explaining the generation of individual sentences. The generative story is formally presented in Figure 2.

We associate with each semantic class c, $c = 1, 2, \dots,$ a distribution of its syntactic realizations ϕ_c. For example, for the frame WinPrize illustrated in Figure 1 this distribution would concentrate at syntactic fragments corresponding to lexical items *"defeated"*, *"secured the victory"* and *"won"*. The distribution is drawn from $DP(w^{(C)}, H^{(C)})$, where $H^{(C)}$ is a base measure over syntactic subtrees. We use a simple generative process to define the probability of a subtree, the underlying model is similar to the base measures used in the Bayesian tree-substitution grammars (Cohn et al., 2009). We start by generating a word w uniformly from the treebank distribution, then we decide on the number of dependents of w using the geometric distribution $Geom(q^{(C)})$. For every dependent we generate a dependency relation r and a lexical form w' from $P(r|w)P(w'|r)$, where probabilities P are based on add-0.1 smoothed treebank counts. The process is continued recursively. The smaller the parameter $q^{(C)}$, the lower is the probability assigned to larger sub-trees.

Parameters $\psi_{c,t}$ and $\psi^{+}_{c,t}$, $t = 1, \dots, T$, define a distribution over vectors $(m_1, m_2, \dots, m_T)$ where m_t is the number of times an argument of type t appears for a given semantic frame occurrence[3]. For the frame WinPrize these parameters would enforce that there exists exactly one Winner and exactly one Opponent for each occurrence of WinPrize. The parameter $\psi_{c,t}$ defines the probability of having at least one argument of type t. If 0 is drawn from $\psi_{c,t}$ then $m_t = 0$, otherwise the number of additional arguments of type t $(m_t - 1)$ is drawn from the geometric distribution $Geom(\psi^{+}_{c,t})$. This generative story is flexible enough to accommodate both argument types which appear at most once per semantic class occurrence (e.g., agents), and argument types which frequently appear multiple times per semantic class occurrence (e.g., arguments corresponding to descriptors).

Parameters $\phi_{c,t}$, $t = 1, \dots, T$, define the dis-

[2] Syntactic realizations are syntactic tree fragments, and therefore they correspond both to syntactic and lexical variations.

[3] For simplicity, we assume that each semantic class has T associated argument types, note that this is not a restrictive assumption as some of the argument types can remain unused, and T can be selected to be sufficiently large to accommodate all important arguments.

```
Parameters:
$\gamma \sim GEM(\alpha_0, \beta_0)$   [top-level proportions of classes]
$\theta_{root} \sim PY(\alpha_{root}, \beta_{root}, \gamma)$   [distrib of sem classes at root]
for each sem class $c = 1, 2, \ldots$:
  $\phi_c \sim DP(w^{(C)}, H^{(C)})$   [distribs of synt realizations]
  for each arg type $t = 1, 2, \ldots T$:
    $\psi_{c,t} \sim Beta(\eta_0, \eta_1)$   [first argument generation]
    $\psi^+_{c,t} \sim Beta(\eta_0^+, \eta_1^+)$   [geom distr for more args]
    $\phi_{c,t} \sim DP(w^{(A)}, H^{(A)})$   [distribs of synt paths]
    $\theta_{c,t} \sim PY(\alpha, \beta, \gamma)$   [distrib of arg fillers]

Data Generation:
for each sentence:
  $c_{root} \sim \theta_{root}$   [choose sem class for root]
  GenSemClass($c_{root}$)

GenSemClass($c$):
  $s \sim \phi_c$   [draw synt realization]
  for each arg type $t = 1, \ldots, T$:
    if $[n \sim \psi_{c,t}] = 1$:   [at least one arg appears]
      GenArgument($c, t$)   [draw one arg]
      while $[n \sim \psi^+_{c,t}] = 1$:   [continue generation]
        GenArgument($c, t$)   [draw more args]

GenArgument($c, t$):
  $a_{c,t} \sim \phi_{c,t}$   [draw synt relation]
  $c'_{c,t} \sim \theta_{c,t}$   [draw sem class for arg]
  GenSemClass($c'_{c,t}$)   [recurse]
```

Figure 2: The generative story for the Bayesian model for unsupervised semantic parsing.

tributions over syntactic paths for the argument type t. In our example, for argument type `Opponent`, this distribution would associate most of the probability mass with relations *pp_over*, *dobj* and *pp_against*. These distributions are drawn from $DP(w^{(A)}, H^{(A)})$. In this paper we only consider paths consisting of a single relation, therefore the base probability distribution $H^{(A)}$ is just normalized frequencies of dependency relations in the treebank.

The crucial part of the model are the selection-preference parameters $\theta_{c,t}$, the distributions of semantic classes c' for each argument type t of class c. For arguments `Winner` and `Opponent` of the frame `WinPrize` these distributions would assign most of the probability mass to semantic classes denoting teams or players. Distributions $\theta_{c,t}$ are drawn from a hierarchical PY process: first, top-level proportions of classes γ are drawn from $GEM(\alpha_0, \beta_0)$, and then the individual distributions $\theta_{c,t}$ over c' are chosen from $PY(\alpha, \beta, \gamma)$.

For each sentence, we first generate a class corresponding to the root of the dependency tree from the root-specific distribution of semantic classes θ_{root}. Then we recursively generate classes for the entire sentence. For a class c, we generate the syntactic realization s and for each of the T types, decide how many arguments of that type to generate (see **GenSemClass** in Figure 2). Then we generate each of the arguments (see **GenArgument**) by first generating a syntactic arc $a_{c,t}$, choosing a class as its filler $c'_{c,t}$ and, finally, recursing.

6 Inference

In our model, latent states, modeled with hierarchical PY processes, correspond to distinct semantic classes and, therefore, their number is expected to be very large for any reasonable model of semantics. As a result, many standard inference techniques, such as Gibbs sampling, or the structured mean-field method are unlikely to result in tractable inference. One of the standard and most efficient samplers for non-hierarchical PY processes are split-merge MH samplers (Jain and Neal, 2000; Dahl, 2003). In this section we explain how split-merge samplers can be applied to our model.

6.1 Split and Merge Moves

On each move, split-merge samplers decide either to merge two states into one (in our case, merge two semantic classes), or split one state into two. These moves can be computed efficiently for our model of semantics. Note that for any reasonable model of semantics only a small subset of the entire set of semantic classes can be used as an argument for some fixed semantic class due to selectional preferences exhibited by predicates. For instance, only teams or players can fill arguments of the frame `WinPrize` in our running example. As a result, only a small number of terms in the joint distribution has to be evaluated on every move we may consider.

When estimating the model, we start with assigning each distinct word (or, more precisely, a tuple of a word's stem and its part-of-speech tag) to an individual semantic class. Then, we would iterate by selecting a random pair of class occurrences, and decide, at random, whether we attempt to perform a split-merge move or a compose-decompose move.

6.2 Compose and Decompose Moves

The compose-decompose operations modify syntactic fragments assigned to semantic classes, composing two neighboring dependency sub-trees or decomposing a dependency sub-tree. If the two randomly-selected syntactic fragments s and s' correspond to different classes, c and c', we attempt to compose them into $\hat{s}$ and create a new semantic class $\hat{c}$. All occurrences of $\hat{s}$ are assigned to this new class $\hat{c}$. For example, if two randomly-selected occurrences have syntactic realizations *"secure"* and *"victory"* they can be composed to obtain the syntactic fragment *"secure* $\xrightarrow{dobj}$ *victory"*. This fragment will be assigned to a new semantic class which can later be merged with other classes, such as the ones containing syntactic realizations *"defeat"* or *"win"*.

Conversely, if both randomly-selected syntactic fragments are already composed in the corresponding class, we attempt to split them.

6.3 Role-Syntax Alignment Move

Merge, compose and decompose moves require recomputation of mapping between argument types (semantic roles) and syntactic fragments. Computing the best statistical mapping is infeasible and proposing a random mapping will result in many attempted moves being rejected. Instead we use a greedy randomized search method called *Gibbs scan* (Dahl, 2003). Though it is a part of the above 3 moves, this alignment move is also used on its own to induce semantic arguments for classes (frames) with a single syntactic realization.

The Gibbs scan procedure is also used during the split move to select one of the newly introduced classes for each considered syntactic fragment.

6.4 Informed Proposals

Since the number of classes is very large, selecting examples at random would result in a relatively low proportion of moves getting accepted, and, consequently, in a slow-mixing Markov chain. Instead of selecting both class occurrences uniformly, we select the first occurrence from a uniform distribution and then use a simple but effective proposal distribution for selecting the second class occurrence.

Let us denote the class corresponding to the first occurrence as c_1 and its syntactic realization as s_1 with a head word w_1. We begin by selecting uniformly randomly whether to attempt a compose-decompose or a split-merge move.

If we chose a compose-decompose move, we look for words (children) which can be attached below the syntactic fragment s_1. We use the normalized counts of these words conditioned on the parent s_1 to select the second word w_2. We then select a random occurrence of w_2; if it is a part of syntactic realization of c_1 then a decompose move is attempted. Otherwise, we try to compose the corresponding clusters together.

If we selected a split-merge move, we use a distribution based on the cosine similarity of lexical contexts of the words. The context is represented as a vector of counts of all pairs of the form (head word, dependency type) and (dependent, dependency type). So, instead of selecting a word occurrence uniformly, each occurrence of every word w_2 is weighted by its similarity to w_1, where the similarity is based on the cosine distance.

As the moves are dependent only on syntactic representations, all the proposal distributions can be computed once at the initialization stage.[4]

7 Empirical Evaluation

We induced a semantic representation over a collection of texts and evaluated it by answering questions about the knowledge contained in the corpus. We used the GENIA corpus (Kim et al., 2003), a dataset of 1999 biomedical abstracts, and a set of questions produced by (Poon and Domingos, 2009). A example question is shown in Figure 3.

All model hyperpriors were set to maximize the posterior, except for $w^{(A)}$ and $w^{(C)}$, which were set to $1.e-10$ and $1.e-35$, respectively. Inference was run for around 300,000 sampling iterations until the percentage of accepted split-merge moves became lower than 0.05%.

Let us examine some of the induced semantic classes (Table 1) before turning to the question answering task. Almost all of the clustered syntactic

[4] In order to minimize memory usage, we used frequency cut-off of 10. For split-merge moves, we select words based on the cosine distance if the distance is below 0.95 and sample the remaining words uniformly. This also reduces the required memory usage.

Class	Variations
1	motif, sequence, regulatory element, response element, element, dna sequence
2	donor, individual, subject
3	important, essential, critical
4	dose, concentration
5	activation, transcriptional activation, transactivation
6	b cell, t lymphocyte, thymocyte, b lymphocyte, t cell, t-cell line, human lymphocyte, t-lymphocyte
7	indicate, reveal, document, suggest, demonstrate
8	augment, abolish, inhibit, convert, cause, abrogate, modulate, block, decrease, reduce, diminish, suppress, up-regulate, impair, reverse, enhance
9	confirm, assess, examine, study, evaluate, test, resolve, determine, investigate
10	nf-kappab, nf-kappa b, nfkappab, nf-kb
11	antiserum, antibody, monoclonal antibody, ab, antisera, mab
12	tnfalpha, tnf-alpha, il-6, tnf

Table 1: Examples of the induced semantic classes.

realizations have a clear semantic connection. Cluster 6, for example, clusters lymphocytes with the exception of thymocyte, a type of cell which generates T cells. Cluster 8 contains verbs roughly corresponding to Cause change of position on a scale frame in FrameNet. Verbs in class 9 are used in the context of providing support for a finding or an action, and many of them are listed as evoking elements for the Evidence frame in FrameNet.

Argument types of the induced classes also show a tendency to correspond to semantic roles. For example, an argument type of class 2 is modeled as a distribution over two argument parts, *prep_of* and *prep_from*. The corresponding arguments define the origin of the cells (*transgenic mouse, smoker, volunteer, donor, . . .*).

We now turn to the QA task and compare our model (**USP-BAYES**) with the results of baselines considered in (Poon and Domingos, 2009). The first set of baselines looks for answers by attempting to match a verb and its argument in the question with the input text. The first version (**KW**) simply returns the rest of the sentence on the other side of the verb, while the second (**KW-SYN**) uses syntactic information to extract the subject or the object of the verb.

	Total	Correct	Accuracy
KW	150	67	45%
KW-SYN	87	67	77%
TR-EXACT	29	23	79%
TR-SUB	152	81	53%
RS-EXACT	53	24	45%
RS-SUB	196	81	41%
DIRT	159	94	59%
USP-MLN	334	295	88%
USP-BAYES	325	259	80%

Table 2: Performance on the QA task.

Other baselines are based on state-of-the-art relation extraction systems. When the extracted relation and one of the arguments match those in a given question, the second argument is returned as an answer. The systems include TextRunner (**TR**) (Banko et al., 2007), RESOLVER (**RS**) (Yates and Etzioni, 2009) and **DIRT** (Lin and Pantel, 2001). The **EXACT** versions of the methods return answers when they match the question argument exactly, and the **SUB** versions produce answers containing the question argument as a substring.

Similarly to the MLN system (**USP-MLN**), we generate answers as follows. We use our trained model to parse a question, i.e. recursively decompose it into lexical items and assign them to semantic classes induced at training. Using this semantic representation, we look for the type of an argument missing in the question, which, if found, is reported as an answer. It is clear that overly coarse clusters of argument fillers or clustering of semantically related but not equivalent relations can hurt precision for this evaluation method.

Each system is evaluated by counting the answers it generates, and computing the accuracy of those answers.[5] Table 2 summarizes the results. First, both USP models significantly outperform all other baselines: even though the accuracy of KW-SYN and TR-EXACT are comparable with our accuracy, the number of correct answers returned by USP-Bayes is 4 and 11 times smaller than those of KW-SYN and TR-EXACT, respectively. While we are not beating the MLN baseline, the difference is not significant. The effective number of questions is relatively small (less than 80 different questions are answered by any of the models). More than 50% of USP-BAYES mistakes were due to wrong interpretation of only 5 different questions. From another point of view, most of the mistakes are explained

[5] The true recall is not known, as computing it would require exhaustive annotation of the entire corpus.

Question: *What does cyclosporin A suppress?*
Answer: *expression of EGR-2*
Sentence: *As with EGR-3 , expression of EGR-2 was blocked by cyclosporin A .*

Question: *What inhibits tnf-alpha?*
Answer: *IL -10*
Sentence: *Our previous studies in human monocytes have demonstrated that interleukin (IL) -10 inhibits lipopolysaccharide (LPS) -stimulated production of inflammatory cytokines , IL-1 beta , IL-6 , IL-8 , and tumor necrosis factor (TNF) -alpha by blocking gene transcription .*

Figure 3: An example of questions, answers by our model and the corresponding sentences from the dataset.

by overly coarse clustering corresponding to just 3 classes, namely, 30%, 25% and 20% of errors are due to the clusters 6, 8 and 12 (Figure 1), respectively. Though all these clusters have clear semantic interpretation (white blood cells, predicates corresponding to changes and cykotines associated with cancer progression, respectively), they appear to be too coarse for the QA method we use in our experiments. Though it is likely that tuning and different heuristics may result in better scores, we chose not to perform excessive tuning, as the evaluation dataset is fairly small.

8 Related Work

There is a growing body of work on statistical learning for different versions of the semantic parsing problem (e.g., (Gildea and Jurafsky, 2002; Zettlemoyer and Collins, 2005; Ge and Mooney, 2005; Mooney, 2007)), however, most of these methods rely on human annotation, or some weaker forms of supervision (Kate and Mooney, 2007; Liang et al., 2009; Titov and Kozhevnikov, 2010; Clarke et al., 2010) and very little research has considered the unsupervised setting.

In addition to the MLN model (Poon and Domingos, 2009), another unsupervised method has been proposed in (Goldwasser et al., 2011). In that work, the task is to predict a logical formula, and the only supervision used is a lexicon providing a small number of examples for every logical symbol. A form of self-training is then used to bootstrap the model.

Unsupervised semantic role labeling with a generative model has also been considered (Grenager and Manning, 2006), however, they do not attempt to discover frames and deal only with isolated predicates. Another generative model for SRL has been proposed in (Thompson et al., 2003), but the parameters were estimated from fully annotated data.

The unsupervised setting has also been considering for the related problem of learning narrative schemas (Chambers and Jurafsky, 2009). However, their approach is quite different from our Bayesian model as it relies on similarity functions.

Though in this work we focus solely on the unsupervised setting, there has been some successful work on semi-supervised semantic-role labeling, including the Framenet version of the problem (Fürstenau and Lapata, 2009). Their method exploits graph alignments between labeled and unlabeled examples, and, therefore, crucially relies on the availability of labeled examples.

9 Conclusions and Future Work

In this work, we introduced a non-parametric Bayesian model for the semantic parsing problem based on the hierarchical Pitman-Yor process. The model defines a generative story for recursive generation of lexical items, syntactic and semantic structures. We extend the split-merge MH sampling algorithm to include composition-decomposition moves, and exploit the properties of our task to make it efficient in the hierarchical setting we consider.

We plan to explore at least two directions in our future work. First, we would like to relax some of unrealistic assumptions made in our model: for example, proper modeling of alterations requires joint generation of syntactic realizations for predicate-argument relations (Grenager and Manning, 2006; Lang and Lapata, 2010), similarly, proper modeling of nominalization implies support of arguments not immediately local in the syntactic structure. The second general direction is the use of the unsupervised methods we propose to expand the coverage of existing semantic resources, which typically require substantial human effort to produce.

Acknowledgements

The authors acknowledge the support of the MMCI Cluster of Excellence, and thank Chris Callison-Burch, Alexis Palmer, Caroline Sporleder, Ben Van Durme and the anonymous reviewers for their helpful comments and suggestions.

References

O. Abend, R. Reichart, and A. Rappoport. 2009. Unsupervised argument identification for semantic role labeling. In *Proceedings of ACL-IJCNLP*, pages 28–36, Singapore.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2670–2676.

Matthew J. Beal, Zoubin Ghahramani, and Carl E. Rasmussen. 2002. The infinite hidden markov model. In *Machine Learning*, pages 29–245. MIT Press.

David Blackwell and James B. MacQueen. 1973. Ferguson distributions via polya urn schemes. *The Annals of Statistics*, 1(2):353–355.

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the CoNLL-2005 Shared Task: Semantic Role Labeling. In *Proceedings of the 9th Conference on Natural Language Learning, CoNLL-2005*, Ann Arbor, MI USA.

Nathanael Chambers and Dan Jurafsky. 2009. Unsupervised learning of narrative schemas and their participants. In *Proc. of the Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP)*.

James Clarke, Dan Goldwasser, Ming-Wei Chang, and Dan Roth. 2010. Driving semantic parsing from the world's response. In *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*.

Trevor Cohn, Sharon Goldwater, and Phil Blunsom. 2009. Inducing compact but accurate tree-substitution grammars. In *HLT-NAACL*, pages 548–556.

David B. Dahl. 2003. An improved merge-split sampler for conjugate dirichlet process mixture models. Technical Report 1086, Department of Statistics, University of Wiscosin - Madison, November.

Jacob Eisenstein, James Clarke, Dan Goldwasser, and Dan Roth. 2009. Reading to learn: Constructing features from semantic abstracts. In *Proceedings of EMNLP*.

Thomas S. Ferguson. 1973. A bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230.

C. J. Fillmore, C. R. Johnson, and M. R. L. Petruck. 2003. Background to framenet. *International Journal of Lexicography*, 16:235–250.

Hagen Fürstenau and Mirella Lapata. 2009. Graph alignment for semi-supervised semantic role labeling. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

Ruifang Ge and Raymond J. Mooney. 2005. A statistical semantic parser that integrates syntax and semantics. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CONLL-05)*, Ann Arbor, Michigan.

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

Dan Goldwasser, Roi Reichart, James Clarke, and Dan Roth. 2011. Confidence driven unsupervised semantic parsing. In *Proc. of the Meeting of Association for Computational Linguistics (ACL)*, Portland, OR, USA.

Trond Grenager and Christoph Manning. 2006. Unsupervised discovery of a statistical verb lexicon. In *Proceedings of Empirical Methods in Natural Language Processing (EMNLP)*.

Sonia Jain and Radford Neal. 2000. A split-merge markov chain monte carlo procedure for the dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13:158–182.

Mark Johnson, Thomas L. Griffiths, and Sharon Goldwater. 2007. Bayesian inference for PCFGs via Markov chain Monte Carlo. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, Rochester, USA.

Rohit J. Kate and Raymond J. Mooney. 2007. Learning language semantics from ambigous supervision. In *Association for the Advancement of Artificial Intelligence (AAAI)*, pages 895–900.

Jin-Dong Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19:i180–i182.

Joel Lang and Mirella Lapata. 2010. Unsupervised induction of semantic roles. In *Proceedings of the 48rd Annual Meeting of the Association for Computational Linguistics (ACL)*, Uppsala, Sweden.

Percy Liang, Slav Petrov, Michael Jordan, and Dan Klein. 2007. The infinite PCFG using hierarchical dirichlet processes. In *Joint Conf. on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 688–697, Prague, Czech Republic.

Percy Liang, Michael I. Jordan, and Dan Klein. 2009. Learning semantic correspondences with less supervision. In *Proc. of the Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing (ACL-IJCNLP)*.

Dekang Lin and Patrick Pantel. 2001. Dirt – discovery of inference rules from text. In *Proc. of International Conference on Knowledge Discovery and Data Mining*, pages 323–328.

Raymond J. Mooney. 2007. Learning for semantic parsing. In *Proceedings of the 8th International Conference on Computational Linguistics and Intelligent Text Processing*, pages 982–991.

Alexis Palmer and Caroline Sporleder. 2010. Evaluating framenet-style semantic parsing: the role of coverage gaps in framenet. In *Proceedings of the Conference on Computational Linguistics (COLING-2000)*, Beijing.

Jim Pitman. 2002. Poisson-dirichlet and gem invariant distributions for split-and-merge transformations of an interval partition. *Combinatorics, Probability and Computing*, 11:501–514.

Hoifung Poon and Pedro Domingos. 2009. Unsupervised semantic parsing. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, (EMNLP-09)*.

Matt Richardson and Pedro Domingos. 2006. Markov logic networks. *Machine Learning*, 62:107–136.

Alan Ritter, Mausam, and Oren Etzioni. 2010. A latent dirichlet allocation method for selectional preferences. In *Proceedings of the 48rd Annual Meeting of the Association for Computational Linguistics (ACL)*, Uppsala, Sweden.

Diarmuid Ó Séaghdha. 2010. Latent variable models of selectional preference. In *Proceedings of the 48rd Annual Meeting of the Association for Computational Linguistics (ACL)*, Uppsala, Sweden.

R. Swier and S. Stevenson. 2004. Unsupervised semantic role labelling. In *Proceedings of EMNLP*, pages 95–102, Barcelona, Spain.

Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Y. W. Teh. 2006. A hierarchical Bayesian language model based on Pitman-Yor processes. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 985–992.

Cynthia A. Thompson, Roger Levy, and Christopher D. Manning. 2003. A generative model for semantic role labeling. In *In Senseval-3*, pages 397–408.

Ivan Titov and Mikhail Kozhevnikov. 2010. Bootstrapping semantic analyzers from non-contradictory texts. In *Proceedings of the 48rd Annual Meeting of the Association for Computational Linguistics (ACL)*, Uppsala, Sweden.

Alexander Yates and Oren Etzioni. 2009. Unsupervised methods for determining object and relation synonyms on the web. *Journal of Artificial Intelligence Research*, 34:255–296.

B. Zapirain, E. Agirre, L. L. Màrquez, and M. Surdeanu. 2010. Improving semantic role classification with selectional prefrences. In *Proceedings of the Meeting of the North American chapter of the Association for Computational Linguistics (NAACL 2010)*, Los Angeles.

Luke Zettlemoyer and Michael Collins. 2005. Learning to map sentences to logical form: Structured classification with probabilistic categorial grammar. In *Proceedings of the Twenty-first Conference on Uncertainty in Artificial Intelligence*, Edinburgh, UK, August.