Embedded Semantic Lexicon Induction with Joint Global and Local Optimization

Sujay Kumar Jauhar
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA, USA
sjauhar@cs.cmu.edu

Eduard Hovy
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA, USA
hovy@cs.cmu.edu

Abstract

Creating annotated frame lexicons such as PropBank and FrameNet is expensive and labor intensive. We present a method to induce an embedded frame lexicon in an minimally supervised fashion using nothing more than unlabeled predicate-argument word pairs. We hypothesize that aggregating such pair selectional preferences across training leads us to a global understanding that captures predicate-argument frame structure. Our approach revolves around a novel integration between a predictive embedding model and an Indian Buffet Process posterior regularizer. We show, through our experimental evaluation, that we outperform baselines on two tasks and can learn an embedded frame lexicon that is able to capture some interesting generalities in relation to hand-crafted semantic frames.

1 Introduction

Semantic lexicons such as PropBank (Palmer et al., 2005) and FrameNet (Baker et al., 1998) contain information about predicate-argument frame structure. These frames capture knowledge about the affinity of predicates for certain types of arguments, their number and their semantic nature, regardless of syntactic realization.

For example, PropBank specifies frames in the following manner:

- eat → $[\text{agent}]_0$, $[\text{patient}]_1$
- give → $[\text{agent}]_0$, $[\text{theme}]_1$, $[\text{recipient}]_2$

These frames provide semantic information such as the fact that "eat" is transitive, while "give" is ditransitive, or that the beneficiary of one action is a "patient", while the other is a "recipient".

This structural knowledge is crucial for a number of NLP applications. Information about frames has been successfully used to drive and improve diverse tasks such as information extraction (Surdeanu et al., 2003), semantic parsing (Das et al., 2010) and question answering (Shen and Lapata, 2007), among others.

However, building these frame lexicons is very expensive and time consuming. Thus, it remains difficult to port applications from resource-rich languages or domains to data impoverished ones. The NLP community has tackled this issue along two different lines of unsupervised work.

At the local token level, researchers have attempted to model frame structure by the selectional preference of predicates for certain arguments (Resnik, 1997; Séaghdha, 2010). For example, on this problem a good model might assign a high probability to the word "pasta" occurring as an argument of the word "eat".

Contrastingly, at the global type level, work has focussed on inducing frames by clustering predicates and arguments in a joint framework (Lang and Lapata, 2011a; Titov and Klementiev, 2012b). In this case, one is interested in associating predicates such as "eat", "consume", "devour", with a joint clustering of arguments such as "pasta", "chicken", "burger".

While these methods have been useful for several problems, they also have shortcomings. Selectional preference modelling only captures local predicate-argument affinities, but does not aggregate these associations to arrive at a structural understanding of frames.

Meanwhile, frame induction performs clustering at a global level. But most approaches tend to be algorithmic methods (or some extension thereof) that focus on semantic role labelling.

*Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 209–219,
Vancouver, Canada, August 3-4, 2017. ©2017 Association for Computational Linguistics

Their lack of portable features or model parameters unfortunately means they cannot be used to solve other applications or problems that require lexicon-level information – such as information extraction or machine translation. Another limitation is that they always depend on high-level linguistic annotation, such as syntactic dependencies, which may not exist in resource-poor settings.

Thus, in this paper we propose to combine the two approaches to induce a frame semantic lexicon in a minimally supervised fashion with nothing more than unlabeled predicate-argument word pairs. Additionally, we will learn an embedded lexicon that jointly produces embeddings for predicates, arguments and an automatically induced collection of latent slots. The embeddings provide flexibility for usage in downstream applications, where predicate-argument affinities can be computed at will.

To jointly capture the local and global streams of knowledge we propose a novel integration between a predictive embedding model and the posterior of an Indian Buffet Process. The embedding model maximizes the predictive accuracy of predicate-argument selectional preference at the local token level, while the posterior of the Indian Buffet process induces an optimal set of latent slots at the global type level that capture the regularities in the learned predicate embeddings.

We evaluate our approach and show that our models are able to outperform baselines on both the local and global level of frame knowledge. At the local level we score higher than a standard predictive embedding model on selectional preference, while at the global level we outperform a syntactic baseline on lexicon overlap with PropBank. Finally, our analysis on the induced latent slots yields insight into some interesting generalities that we are able to capture from unlabeled predicate-argument pairs.

2 Related Work

The work in this paper relates to research on identifying predicate-argument structure in both local and global contexts. These related areas of research correspond to the NLP community's work respectively on selectional preference modelling and semantic frame induction (which is also known variously as unsupervised semantic role labelling or role induction).

Selectional preference modelling seeks to capture the semantic preference of predicates for certain arguments in local contexts. These preferences are useful for many tasks, including unsupervised semantic role labelling (Gildea and Jurafsky, 2002) among others.

Previous work has sought to acquire these preferences using various means, including ontological resources such as WordNet (Resnik, 1997; Ciaramita and Johnson, 2000), latent variable models (Rooth et al., 1999; Séaghdha, 2010; Ritter et al., 2010) and distributional similarity metrics (Erk, 2007). Most closely related to our contribution is the work by Van de Cruys (2014) who use a predictive neural network to capture predicate-argument associations.

To the best of our knowledge, our research is the first to attempt using selectional preference as a basis for directly inducing semantic frames.

At the global level, frame induction subsumes selectional preference by attempting to group arguments of predicates into coherent and cohesive clusters. While work in this area has included diverse approaches, such as leveraging example-based representations (Kawahara et al., 2014) and cross-lingual resources (Fung and Chen, 2004; Titov and Klementiev, 2012b), most attempts have focussed on two broad categories. These are latent variable driven models (Grenager and Manning, 2006; Cheung et al., 2013) and similarity driven clustering models (Lang and Lapata, 2011a,b),

Our work includes elements of both major categories, since we use latent slots to represent arguments, but an Indian Buffet process induces these latent slots in the first place. The work of Titov and Klementiev (2012a) and Woodsend and Lapata (2015) are particularly relevant to our research. The former use another non-parametric Bayesian model (a Chinese Restaurant process) in their work, while the latter embed predicate-argument structures before performing clustering.

Crucially, however all these previous efforts induce frames that are not easily portable to applications other than semantic role labelling (for which they are devised). Moreover, they rely on syntactic cues to featurize and help cluster argument instances. To the best of our knowledge, ours is the first attempt to go from unlabeled bag-of-arguments to induced frame embeddings without any reliance on annotated data.

3 Joint Local and Global Frame Lexicon Induction

In this section we present our approach to induce a frame lexicon with latent slots. Following prior work on frame induction (Lang and Lapata, 2011a; Titov and Klementiev, 2012a), the procedural pipeline can be split into two distinct phases: argument identification and argument clustering.

As with previous work, we focus on the latter stage, and assume that we have unlabeled predicate-argument structure pairs – given to us from gold standard annotation or through heuristic means (Lang and Lapata, 2014).

We begin with preliminary notation. Given a vocabulary of predicate types $P = \{p_1, ..., p_n\}$ and contextual argument types $A = \{a_1, ..., a_m\}$. Let $C = \{(p_1, a_1), ..., (p_N, a_N)\}$ be a corpus of predicate-argument word token pairs[1]. Given this corpus, we will attempt to learn an optimal set of model parameters θ that maximizes a regularized likelihood over the corpus.

The model parameters include $V = \{v_i \mid \forall p_i \in P\}$ an $n \times d$ embedding matrix for the predicates and $U = \{u_i \mid \forall a_i \in A\}$ an $m \times d$ embedding matrix for the arguments. Additionally, assuming K latent frame slots we define $Z = \{z_{ik}\}$ an $n \times k$ binary matrix that represents the presence or absence of the slot k for the predicate i, and a latent $K \times d$ weight matrix $S = \{s_k \mid 1 \leq k \leq K\}$ that associates a weight vector to each latent slot.

The generalized form of the objective we optimize is given by:

$$\hat{\theta} = \arg\max_{\theta} \sum_{(p_i,a_i)\in C} \log \left(\sum_k Pr(a_i|p_i, z_{ik}, s_k) \right) + \log pr_{\theta}(Z|V) \quad (1)$$

This objective has two parts: a likelihood term, and a posterior regularizer. The former will be responsible for modelling the predictive accuracy of selectional-preference at a local level, while the latter will capture global consistencies for an optimal set of latent slots.

We detail the parametrization of each of these components separately in what follows.

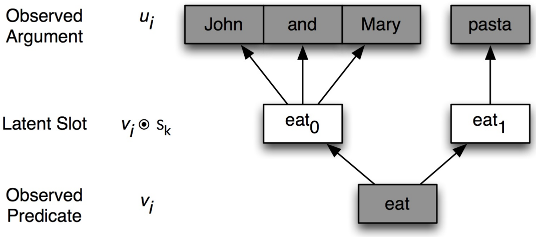

Figure 1: The generative story depicting the realization of an argument from a predicate. Argument words are generated from latent argument slots. Observed variables are shaded in grey, while latent variables are in white.

3.1 Local Predicate-Argument Likelihood

The likelihood term of our model is based on the popular Skip-gram model from Mikolov et al. (2013) but suitably extended to incorporate the latent frame slots and their associated weights. Specifically, we define the probability for a single predicate-argument pair (p_i, a_i) as:

$$Pr(a_i|p_i) = \sum_k Pr(a_i|p_i, z_{ik}, s_k) = \sum_k z_{ik} \frac{\exp((v_i \odot s_k) \cdot u_i)}{\sum_{a_{i'}} \exp((v_i \odot s_k) \cdot u_{i'})} \quad (2)$$

where $\odot$ represents the element-wise multiplication operator. Intuitively, in the likelihood term we weight a general predicate embedding to a slot-specific representations, which then predicts a specific argument. This is graphically represented in Figure 1.

3.2 Global Latent Slot Regularization

The posterior regularization term in equation 1 seeks to balance the likelihood term by yielding an optimal set of latent slots, given the embedding matrix of predicates.

We choose the posterior of an Indian Buffet process (IBP) (Griffiths and Ghahramani, 2005) in this step to induce an optimal latent binary matrix Z. The IBP itself places a prior on equivalence classes of infinite dimensional sparse binary matrices, and is the infinite limit ($K \to \infty$) of a beta-Bernoulli model.

$$\begin{aligned} \pi_k &\sim Beta(\alpha/K, 1) \\ z_{ik} &\sim Bernoulli(\pi_k) \end{aligned} \quad (3)$$

Given a suitable likelihood function and some

[1] In this work, we assume argument chunks are broken down into individual words, – to increase training data size – but the model remains agnostic to this decision.

data, inference in an IBP computes a posterior that yields an optimal *finite* binary matrix with respect to regularities in the data.

Setting the *data*, in our case, to be the embedding matrix of predicates V, this gives us precisely what we are seeking. It allows us to find regularities in the embeddings, while factorizing them according to these consistencies. The model also automatically optimizes the number of and relationship between latent slots, rather than setting these a priori.

Other desiderata are encoded as well, including the fact that the the matrix Z remains sparse, while the frequency of slots follows a power-law distribution proportional to Poisson(α). In practise, this captures the power-law distribution of relational slots in real-world semantic lexicons such as PropBank (Palmer et al., 2005). All of these properties stem directly from the choice of prior, and are a natural consequence of using an IBP.

In this paper, we use a linear-Gaussian model as the likelihood function. This is a popular model that has been applied to several problems, and for which different approximate inference strategies have been developed (Doshi-Velez et al., 2009; Doshi-Velez and Ghahramani, 2009). According to his model, the predicate embeddings are distributed as:

$$v_i \sim Gaussian(z_i W, \sigma_V^2 \mathbf{I}) \tag{4}$$

where W is a $K \times d$ matrix of weights and σ_V is a hyperparameter.

For a detailed derivation of the posterior of an IBP prior with a linear-Gaussian likelihood, we point the reader to Griffiths and Ghahramani (2011), who provide a meticulous summary.

3.3 Optimization

Since our objective in equation 1 contains two distinct components, we can optimize using alternating maximization. Although guaranteed convergence for this technique only exist for convex functions, it has proven successful even for non-convex problems (Jain et al., 2013; Netrapalli et al., 2013).

We thus alternate between keeping Z fixed and optimizing the parameters V, U, S in the likelihood component of section 3.1, and keeping V fixed and optimizing the parameters Z in the posterior regularization component of section 3.2.

In practise, the likelihood component is optimized using negative sampling with EM for the latent slots. In particular we use *hard* EM, to select a single slot before taking gradient steps with respect to the model parameters. This was shown to work well for Skip-gram style models with latent variables by Jauhar et al. (2015).

In the E-Step we find the best latent slot for a particular predicate-argument pair:

$$\hat{k} = \arg\max_k Pr(a_i | p_i, z_{ik}, s_k) \tag{5}$$

We follow this by making stochastic gradient updates to the model parameters U, V, S in the M-Step using the negative sampling objective:

$$\log z_{i\hat{k}} \sigma\left((v_i \odot s_{\hat{k}}) \cdot u_i\right) + \sum_l \mathbb{E}_{a_{i'} \sim Pr_n(a)} \left[\log z_{i\hat{k}} \sigma\left((v_i \odot s_{\hat{k}}) \cdot u_{i'}\right)\right] \tag{6}$$

where $\sigma(\cdot)$ is the sigmoid function, $Pr_n(a)$ is a unigram noise distribution over argument types and l is the negative sampling parameter.

As for optimizing the posterior regularization component, an approximate inference technique such as Gibbs sampling must be used. In Gibbs sampling we iteratively sample individual z_{ik} terms from the posterior:

$$Pr(z_{ik} | X, Z_{-ik}) \propto Pr(X|Z) \cdot Pr(z_{ik} | Z_{-ik}) \tag{7}$$

where Z_{-ik} is the Markov blanket of z_{ik} in Z. The prior and likelihood terms are respectively those of equations 3 and 4. Doshi-Velez and Ghahramani (2009) present an accelerated version of Gibbs sampling for this model, that computes the likelihood and prior terms efficiently. We use this approach in our work since it has the benefits of mixing like a collapsed sampler, while maintaining the running time of an uncollapsed sampler.

In conclusion, the optimization steps iteratively refine the parameters V, U, S to be better predictors of the corpus, while Z is updated to best factorize the regularities in the predicate embeddings V, thereby capturing better relational slots.

3.4 Relational Variant

In addition to the standard model introduced above, we also experiment with an extension where the input corpus consists of predicate-argument-relation triples instead of just predicate-argument pairs. These relations are observed relations, and should not be confused with the latent slots of the model.

To accommodate this change we modify the argument embedding matrix U to be of dimensions $m \times \frac{d}{2}$ and introduce a new $q \times \frac{d}{2}$ embedding matrix $R = \{r_i \mid 1 \leq i \leq q\}$ for the q observed relation types.

Then, wherever the original model calls for an argument vector u_i (which had dimensionality d) we instead replace it with a concatenated argument-relation vector $[u_i; r_j]$ (which now also has dimensionality d). During training, we must make gradient updates to R in addition to all the other model parameters as usual.

While this relation indicator can be used to capture arbitrary relational information, in this paper we set it to a combination of the directionality of the argument with respect to the predicate (L or R), and the preposition immediately preceding the argument phrase (or *None* if there isn't one). Thus, for example, we have relational indicators such as "L-on", "R-before", "L-because", "R-None", etc. We obtain a total of 146 such relations.

Note, that in keeping with the goals of this work, these relation indicators still require no annotation (prepositions are closed-class words than can be enumerated).

4 Experiments and Evaluation

In what follows, we detail experimental results on two quantitative evaluation tasks: at the local and global levels of predicate-argument structure. In particular we evaluate on pseudo disambiguation of selectional preference, and semantic frame lexicon overlap. We also qualitatively inspect the learned latent relations against hand-annotated roles. We first specify the implementational details.

4.1 Implementational Details

We begin by pre-training standard skip-gram vectors (Mikolov et al., 2013) on the NY-Times section of the Gigaword corpus, which consists of approximately 1.67 billion word tokens. These vectors are used as initialization for the embedding matrices V and U, before our iterative optimization. While this step is not strictly required, we found that it leads to generally better results than random initialization given the relatively small size of our predicate-argument training corpus.

For training our models, we use a combination of the training data released for the CoNLL 2008 shared task (Surdeanu et al., 2008) and the extended PropBank release which covers annotations of the Ontonotes (Hovy et al., 2006) and English Web Treebank (Bies et al., 2012) corpora. We reserve the test portion of the CoNLL 2008 shared task data for one of our evaluations.

In this work, we only focus on verbal predicates. Our training data gives us a vocabulary of 4449 predicates, after pruning verbs that occur fewer than 5 times.

Then, from the training data we extract all predicate-argument pairs using gold standard argument annotations, for the sake of simplicity. Note that previous unsupervised frame induction work also uses gold argument mentions (Lang and Lapata, 2011a; Titov and Klementiev, 2012b). Our method, however, does not depend on this, or any other annotation, and we could as easily use the output from an automated system such as Abend et al. (2009) instead.

In this manner, we obtain a total of approximately 3.35 million predicate-argument word pairs on which to train.

Using this data we train a total of 4 distinct models: a base model and a relational variant (see Section 3.4), both of which are trained with two different IBP hyperparameters of $\alpha = 0.35$ and $\alpha = 0.7$. The hyperparameter controls the avidity of the model for latent slots (a higher α implies a greater number of induced slots).

This results in the learned number of slots ranging from 17 to 30, with the conservative model averaging about 4 latent slots per word, while the permissive model averaging about 6 latent slots per word.

Since our objective is non-convex we record the training likelihood at each power iteration (including an optimization over both the predictive and IBP components of our objective), and save the model with the highest training likelihood.

We set our embedding size to $d = 100$ and, after training, obtain latent slot factors ranging in number from 15 to 30.

Model	α	Variant	k slots	% Acc
Skip-gram	-	-	-	0.77
pa2IBPVec	0.35	Standard	17	0.81
		Relational	15	**0.84**
	0.7	Standard	27	0.81
		Relational	30	0.81

Table 1: Results on pseudo disambiguation of selectional preference. Numbers are in % accuracy of distinguishing true arguments from false ones. Our models all outperform the skip-gram baseline.

4.2 Pseudo Disambiguation of Selection Preference

The pseudo disambiguation task aims to evaluate our models' ability to capture predicate-argument knowledge at the local level. In this task, systems are presented with a set of triples: a predicate, a true argument and a fake argument. The systems are evaluated on the percentage of true arguments they are able to select.

For example, given a triple:

resign, post, liquidation

a successful model should rate the pair "resign-post" higher than "resign-liquidation".

This task has often been used in the selectional preference modelling literature as a benchmark task (Rooth et al., 1999; Van de Cruys, 2014) .

To obtain the triples for this task we use the test set of the CoNLL 2008 shared task data. In particular, for every verbal predicate mention in the data we select a random nominal word from each of its arguments phrase chunks to obtain a true predicate-argument word pair. Then, to introduce distractors, we sample a random nominal from a unigram noise distribution. In this way we obtain 9859 pseudo disambiguation triples as our test set.

We use our models to score a word pair by taking the probability of the pair under our model, using the best latent slot:

$$\max_k z_{ik} \sigma \left((v_i \odot s_k) \cdot u_i \right) \tag{8}$$

where v_i and u_i are predicate and argument embeddings respectively, z_{ik} is the binary indicator of the k'th slot for the i'th predicate, and s_k is the slot specific weight vector. The argument in the higher scoring pair is selected as the correct one.

In the relational variant, instead of the single argument vector u_i we also take a max over the relation indicators – since the exact indicator is not observed at test time.

We compare our models against a standard skip-gram model (Mikolov et al., 2013) trained on the same data. Word pairs in this model are scored using the dot product between their associated skip-gram vectors.

This is a fair comparison since our models as well as the skip-gram model have access to the same data – namely predicates and their neighboring argument words. They are trained on their ability to discriminate true argument words from randomly sampled noise. The evaluation then, is whether the additionally learned slot structure helps in differentiating true arguments from noise. The results of this evaluation are presented in Table 1.

The results show that all our models outperform the skip-gram baseline. This demonstrates that the added structural information gained from latent slots in fact help our models to better capture predicate-argument affinities in local contexts.

The impact of latent slots or additional relation information does not seem to impact basic performance, however. This could be because of the trade-off that occurs when a more complex model is learned from the same amount of limited data.

4.3 Frame Lexicon Overlap

Next, we evaluate our models at their ability to capture global predicate-argument structure. Previous work on frame induction has focussed on evaluating instance-based argument overlap with gold standard annotations in the context of semantic role labelling (SRL). Unfortunately, because our models operate on individual predicate-argument words rather than argument spans a fair comparison becomes problematic.

But unlike previous work, which clusters argument instances, our approach produces a model as a result of training. We can thus directly evaluate this model's latent slot factors against a gold standard frame lexicon. Our evaluation framework is, in many ways based on the metrics used in unsupervised SRL, except applied at the "type" lexicon level rather than the corpus-based "token" cluster level.

In particular, given a gold frame lexicon Ω with K^* real argument slots (i.e. the total number of

Model	α	Variant	Coarse			Fine		
			PU	CO	F1	PU	CO	F1
Syntax	-	-	0.71	0.87	0.78	0.70	0.91	0.79
pa2IBPVec	0.35	Standard	0.76	0.89	0.82	0.76	0.97	0.85
		Relational	0.73	0.90	0.81	0.73	0.97	0.83
	0.7	Standard	0.79	0.91	**0.85**	0.79	**0.98**	0.87
		Relational	**0.80**	**0.92**	**0.85**	**0.80**	**0.98**	**0.88**

Table 2: Results on the lexicon overlap task. Our models outperform the syntactic baseline on all the metrics.

possible humanly assigned arguments in the lexicon), we evaluate our models' latent slot matrix Z in terms of its overlap with the gold lexicon.

We define *purity* as the average proportion of overlap between predicted latent slots and their maximally similar gold lexicon slots:

$$PU = \frac{1}{K}\sum_{k} \max_{k'} \frac{1}{n} \sum_{i} \delta(\omega_{ik'}, z_{ik}) \qquad (9)$$

where $\delta(\cdot)$ is an indicator function. Given that the ω's and z's we compare are binary values, this indicator function is effectively an "XNOR" gate.

Similarly we define *collocation* as the average proportion of overlap between gold standard slots and their maximally similar predicted latent slots:

$$CO = \frac{1}{K^{*}}\sum_{k'} \max_{k} \frac{1}{n} \sum_{i} \delta(\omega_{ik'}, z_{ik}) \qquad (10)$$

Given, the *purity* and *collocation* metrics we can define the $F1$ score as the harmonic mean of the two:

$$F1 = \frac{2 \cdot CO \cdot PU}{CO + PU} \qquad (11)$$

In our experiments we use the frame files provided with the PropBank corpus (Palmer et al., 2005) as gold standard. We derive two variants from the frame files.

The first is a coarse-grained lexicon. In this case, we extract only the functional arguments of verbs in our vocabulary as gold standard slots. These functions correspond to broad semantic argument types such as "prototypical agent", "prototypical patient", "instrument", "benefactive", etc. A total of 16 gold slots are produced in this manner, and are mapped to indices. For every verb the corresponding binary ω vector marks the existence or not of the different functional arguments according to the gold frame files.

The second variant is a fine-grained lexicon. Here, in addition to functional arguments we also consider the numerical argument with which it is associated, such as "ARG0", "ARG1" etc. Note that a single functional argument may appear with more than one numerical slot with different verbs over the entire lexicon. The fine-grained lexicon yields 72 gold slots.

We compare our models against a baseline inspired from the syntactic baseline often used for evaluating unsupervised SRL models. For unsupervised SRL, syntax has proven to be a difficult to outperform baseline (Lang and Lapata, 2014).

This baseline is constructed by taking the 21 most frequent syntactic labels in the training data and associating them each with a slot. All other syntactic labels are associated with a 22nd generic slot. Given these slots, we associate a verbal predicate with a specific slot if it takes on the corresponding syntactic argument in the training data. The results on the lexicon overlap task are presented in Table 2.

They show that our models consistently outperform the syntactic baseline on all metrics in both the coarse-grained and fine-grained settings. We conclude that our models are better able to capture predicate-argument structure at a global level.

Inspecting and comparing the results of our different models seems to indicate that we perform better when our IBP posterior allows for a greater number of latent slots. This happens when the hyperparameter $\alpha = 0.7$.

Additionally our models consistently perform better on the fine-grained lexicon than on the coarse-grained one. The former itself does not necessarily represent an easier benchmark, since there is hardly any difference in the $F1$ score of the syntactic baseline on the two lexicons.

Overall it would seem that allowing for a greater number of latent slots does help capture global

Predicate	Latent Slot							
	1	2	3	5	6	8	10	12
provide	A0			A1	A2			A2
enter	A0	A1						AM-ADV
praise	A0	A1				A2		
travel	A0	A0				AM-PNC	AM-TMP	
distract	A0		A1	A2				
overcome	AM-TMP		A0	A0				

Table 3: Examples for several predicates with mappings of latent slots to the majority class of the closest argument vector in the shared embedded space.

predicate-argument structure better. This makes sense, if we consider the fact that we are effectively trying to factorize a dense representation (the predicate embeddings) with IBP inference. Thus allowing for a greater number of latent factors permits the discovery of greater structural consistency within these embeddings.

This finding does have some problematic implications, however. Increasing the IBP hyperparameter α arbitrarily represents a computational bottleneck since inference scales quadratically with the number of latent slots K. There is also the problem of splitting argument slots too finely, which may result in optimizing purity at the expense of collocation. A solution to this trade-off between performance and inference time remains for future work.

4.4 Qualitative Analysis of Latent Slots

To better understand the nature of the latent slots induced by our model we conduct an additional qualitative analysis. The goal of this analysis is to inspect the kinds of generalities about semantic roles that our model is able to capture from completely unannotated data.

Table 3 lists some examples of predicates and their associated latent slots. The latent slots are sorted according to their frequency (i.e. column sum in the binary slot matrix Z). We map each latent slot to the majority semantic role type – from training data – of the closest argument word to the predicate vector in the shared embedding space.

The model for which we perform this qualitative analysis is the standard variant with the IBP hyperparameter set to $\alpha = 0.35$; this model has 17 latent slots. Note that slots that do not feature for any of the verbs are omitted for visual compactness.

There are several interesting trends to notice here. Firstly, the basic argument structure of predicates is often correctly identified, when matched against gold PropBank frame files. For example, the core roles of “enter” identify it as a transitive verb, while “praise”, “provide” and “distract” are correctly shown as ditransitive verbs. Obviously the structure isn’t always perfectly identified, as with the verb “travel” where we are missing both an “ARG1” and an “ARG2”.

In certain cases a single argument type spans multiple slots – as with “A2” for “provide” and “A0” for “travel”. This is not surprising, since there is no binding factor on the model to produce one-to-one mappings with hand-crafted semantic roles. Generally speaking, the slots represent distributions over hand-crafted roles rather than strict mappings. In fact, to expect a one-to-one mapping is unreasonable considering we use no annotations whatsoever.

Nevertheless, there is still some consistency in the mappings. The core arguments of verbs – such as “ARG0” and “ARG1” are typically mapped to the most frequent latent slots. This can be explained by the fact that the more frequent arguments tend to be the ones that are core to a predicate’s frame. This is quite a surprising outcome of the model, considering that it is given no annotation about argument types. Of course, we do not always get this right as can be seen with the case of “overcome”, where a non-core argument occurs in the most frequent slot.

Since this is a data driven approach, we identify non-core roles as well, if they occur with predicates often enough in the data. For example we have the general purpose “AM-ADV” argument of “enter”, and the “ARG-PNC” and “ARG-TMP” (purpose and time arguments) of the verb “travel”. In future work we hope to explore methods that might be able to automatically distinguish core slots from non-core ones.

In conclusion, our model show promise in that it is able to capture some interesting generalities with respect to predicates and their hand-crafted roles, without the need for any annotated data.

5 Conclusion and Future Work

We have presented a first attempt at learning an embedded frame lexicon from data, using no annotated information. Our approach revolves around jointly capturing local predicate-argument affinities with global slot-level consistencies. We model this approach with a novel integration between a predictive embedding model and the posterior of an Indian Buffet Process.

We experiment with our model on two quantitative tasks, each designed to evaluate performance on capturing local and global predicate-argument structure respectively. On both tasks we demonstrate that our models are able to outperform baselines, thus indicating our ability to jointly model the local and global level information of predicate-argument structure.

Additionally, we qualitatively inspect our induced latent slots and show that we are able to capture some interesting generalities with regards to hand-crafted semantic role labels.

There are several avenues of future work we are exploring. Rather than depend on gold argument mentions in training, we hope to fully automate the pipeline to leverage much larger amounts of data. With this greater data size, we also will likely no longer need to break down argument spans into individual words. Instead, we plan to models these spans as chunks using an LSTM.

With this additional modeling power we hope to evaluate on downstream applications such as semantic role labelling, and semantic parsing.

In a separate line of work we hope to be able to parallelize the Indian Buffet Process inference, which remains a bottleneck of our current effort. Speeding up this process will allow us to explore more complex (and potentially better) models.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. This work was supported in part by the following grants: NSF grant IIS-1143703, NSF award IIS-1147810, DARPA grant FA87501220342.

References

Omri Abend, Roi Reichart, and Ari Rappoport. 2009. Unsupervised argument identification for semantic role labeling. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*. Association for Computational Linguistics, pages 28–36.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, pages 86–90.

Ann Bies, Justin Mott, Colin Warner, and Seth Kulick. 2012. English web treebank. *Linguistic Data Consortium, Philadelphia, PA* .

Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. 2013. Probabilistic frame induction. *arXiv preprint arXiv:1302.4813* .

Massimiliano Ciaramita and Mark Johnson. 2000. Explaining away ambiguity: Learning verb selectional preference with bayesian networks. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, pages 187–193.

Dipanjan Das, Nathan Schneider, Desai Chen, and Noah A Smith. 2010. Probabilistic frame-semantic parsing. In *Human language technologies: The 2010 annual conference of the North American chapter of the association for computational linguistics*. Association for Computational Linguistics, pages 948–956.

Finale Doshi-Velez and Zoubin Ghahramani. 2009. Accelerated sampling for the indian buffet process. In *Proceedings of the 26th annual international conference on machine learning*. ACM, pages 273–280.

Finale Doshi-Velez, Kurt T Miller, Jurgen Van Gael, Yee Whye Teh, and Gatsby Unit. 2009. Variational inference for the indian buffet process. In *Proceedings of the Intl. Conf. on Artificial Intelligence and Statistics*. volume 12, pages 137–144.

Katrin Erk. 2007. A simple, similarity-based model for selectional preferences. In *Annual Meeting - Association For Computational Linguistics*. volume 45, page 216.

Pascale Fung and Benfeng Chen. 2004. Biframenet: bilingual frame semantics resource construction by cross-lingual induction. In *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, page 931.

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

Trond Grenager and Christopher D Manning. 2006. Unsupervised discovery of a statistical verb lexicon. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1–8.

Thomas L Griffiths and Zoubin Ghahramani. 2005. Infinite latent feature models and the indian buffet process. In *NIPS*. volume 18, pages 475–482.

Thomas L Griffiths and Zoubin Ghahramani. 2011. The indian buffet process: An introduction and review. *Journal of Machine Learning Research* 12(Apr):1185–1224.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. Ontonotes: the 90% solution. In *Proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers*. Association for Computational Linguistics, pages 57–60.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. 2013. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*. ACM, pages 665–674.

Sujay Kumar Jauhar, Chris Dyer, and Eduard H Hovy. 2015. Ontologically grounded multi-sense representation learning for semantic vector space models. In *HLT-NAACL*. pages 683–693.

Daisuke Kawahara, Daniel Peterson, Octavian Popescu, Martha Palmer, and Fondazione Bruno Kessler. 2014. Inducing example-based semantic frames from a massive amount of verb uses. In *EACL*. pages 58–67.

Joel Lang and Mirella Lapata. 2011a. Unsupervised semantic role induction via split-merge clustering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*. Association for Computational Linguistics, pages 1117–1126.

Joel Lang and Mirella Lapata. 2011b. Unsupervised semantic role induction with graph partitioning. In *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, pages 1320–1331.

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

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Praneeth Netrapalli, Prateek Jain, and Sujay Sanghavi. 2013. Phase retrieval using alternating minimization. In *Advances in Neural Information Processing Systems*. pages 2796–2804.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics* 31(1):71–106.

Philip Resnik. 1997. Selectional preference and sense disambiguation. In *Proceedings of the ACL SIGLEX Workshop on Tagging Text with Lexical Semantics: Why, What, and How*. Washington, DC, pages 52–57.

Alan Ritter, Mausam, and Oren Etzioni. 2010. A latent dirichlet allocation method for selectional preferences. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 424–434.

Mats Rooth, Stefan Riezler, Detlef Prescher, Glenn Carroll, and Franz Beil. 1999. Inducing a semantically annotated lexicon via em-based clustering. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics, pages 104–111.

Diarmuid O Séaghdha. 2010. Latent variable models of selectional preference. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 435–444.

Dan Shen and Mirella Lapata. 2007. Using semantic roles to improve question answering. In *EMNLP-CoNLL*. pages 12–21.

Mihai Surdeanu, Sanda Harabagiu, John Williams, and Paul Aarseth. 2003. Using predicate-argument structures for information extraction. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, pages 8–15.

Mihai Surdeanu, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The conll-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the Twelfth Conference on Computational Natural Language Learning*. Association for Computational Linguistics, pages 159–177.

Ivan Titov and Alexandre Klementiev. 2012a. A bayesian approach to unsupervised semantic role induction. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, pages 12–22.

Ivan Titov and Alexandre Klementiev. 2012b. Crosslingual induction of semantic roles. In *Proceedings of the 50th Annual Meeting of the*

Association for Computational Linguistics: Long Papers-Volume 1. Association for Computational Linguistics, pages 647–656.

Tim Van de Cruys. 2014. A neural network approach to selectional preference acquisition. In *EMNLP*. pages 26–35.

Kristian Woodsend and Mirella Lapata. 2015. Distributed representations for unsupervised semantic role labeling. In *EMNLP*. Citeseer, pages 2482–2491.