Semantic Role Labeling with Neural Network Factors

Nicholas FitzGerald[‡*] **Oscar Täckström**[†] **Kuzman Ganchev**[†] **Dipanjan Das**[†]
[‡]Department of Computer Science and Engineering, University of Washington
[†] Google, New York
nfitz@cs.uw.edu
{oscart,kuzman,dipanjand}@google.com

Abstract

We present a new method for semantic role labeling in which arguments and semantic roles are jointly embedded in a shared vector space for a given predicate. These embeddings belong to a neural network, whose output represents the potential functions of a graphical model designed for the SRL task. We consider both local and structured learning methods and obtain strong results on standard PropBank and FrameNet corpora with a straightforward product-of-experts model. We further show how the model can learn jointly from PropBank and FrameNet annotations to obtain additional improvements on the smaller FrameNet dataset.

1 Introduction

Semantic role labeling (SRL) is the task of identifying the semantic arguments of a predicate and labeling them with their semantic roles. A key challenge in this task is sparsity of labeled data: a given predicate-role instance may only occur a handful of times in the training set. Most existing SRL systems model each semantic role as an atomic unit of meaning, ignoring finer-grained semantic similarity between roles that can be leveraged to share context between similar labels, both within and across annotation conventions.

Low-dimensional embedding representations have been shown to be successful in overcoming sparsity and representing label similarity across a wide range of tasks (Weston et al., 2011; Srikumar and Manning, 2014; Hermann et al., 2014; Lei et al., 2015). In this paper, we present a new model for SRL that embeds candidate arguments and semantic roles (in context of a predicate frame) in a shared vector space. A feed-forward neural network is learned to capture correlations of the respective embedding dimensions to create argument and role representations. The similarity of these two representations, as measured by their dot product, is used to score possible roles for candidate arguments within a graphical model. This graphical model jointly models the assignment of semantic roles to all arguments of a predicate, subject to structural linguistic constraints.

Our model has several advantages. Compared to linear multiclass classifiers used in prior work, vector embeddings of the predictions overcome the assumption of modeling each semantic role as a discrete label, thus capturing fine-grained label similarity. Moreover, since predictions and inputs are embedded in the same vector space, and features extracted from inputs and outputs are decoupled, our approach is amenable to joint learning of multiple annotation conventions, such as PropBank and FrameNet, in a single model. Finally, as with other neural network approaches, our model obviates the need to manually engineer feature conjunctions.

Our underlying inference algorithm for SRL follows Täckström et al. (2015), who presented a dynamic program for structured SRL; it is targeted towards the prediction of full argument spans. Hence, we present empirical results on three span-based SRL datasets: CoNLL 2005 and 2012 data annotated with PropBank conventions, as well as FrameNet 1.5 data. We also evaluate our system on the dependency-based CoNLL 2009 shared task by assuming single word argument spans, that represent semantic dependencies, and limit our experiments to English. On all datasets, our model performs on par with a strong linear model baseline that uses hand-engineered conjunctive features. Due to random parameter initialization and stochasticity in the online learning algorithm used to train our models, we observed considerable variance in performance across datasets. To resolve this variance, we adopt a product-of-experts model that

[*]Work carried out during an internship at Google.

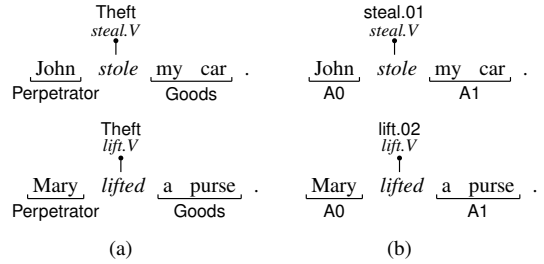


Figure 1: FrameNet (a) and PropBank (b) annotations for two sentences.

combines multiple randomly-initialized instances of our model to achieve state-of-the-art results on the CoNLL 2009 and FrameNet datasets, while coming close to the previous best published results on the other two. Finally, we present even stronger results for FrameNet data (which is scarce) by jointly training the model with PropBank-annotated data.

2 Background

In this section, we briefly describe the SRL task and discuss relevant prior work.

2.1 Semantic Role Labeling

SRL annotations rely on a frame lexicon containing *frames* that could be evoked by one or more *lexical units*. A lexical unit consists of a word lemma conjoined with its coarse-grained part-of-speech tag.[1] Each frame is further associated with a set of possible *core* and *non-core* semantic roles which are used to label its arguments. This description of a frame lexicon covers both PropBank and FrameNet conventions, but there are some differences outlined below. See Figure 1 for example annotations.

PropBank defines frames that are essentially sense distinctions of a given lexical unit. The set of PropBank roles consists of seven generic core roles (labeled A0-A5 and AA) that assume different semantics for different frames, each associating with a subset of the core roles. In addition, there are 21 non-core roles that encapsulate further arguments of a frame, such as temporal (AM-TMP) and locative (AM-LOC) adjuncts. The non-core roles are shared between all frames and assume similar meaning. In contrast, a FrameNet frame often associates with multiple lexical units and the frame lexicon contains several hundred core and non-core roles that are shared across frames. For example, the FrameNet frame Theft could be evoked by the verbs *steal*, *pickpocket*, or *lift*, while PropBank has distinct frames for each of them. The Theft frame also contains the core roles Goods and Perpetrator that additionally belong to the Commercial_transaction and Committing_crime frames respectively.

A typical SRL dataset consists of sentence-level annotations that identify (possibly multiple) target predicates in each sentence, a disambiguated frame for each predicate, and the associated argument spans (or single word argument heads) labeled with their respective semantic roles.

2.2 Related Work

SRL using PropBank conventions (Palmer et al., 2005) has been widely studied. There have been two shared tasks at CoNLL 2004-2005 to identify the phrasal arguments of verbal predicates (Carreras and Màrquez, 2004; Carreras and Màrquez, 2005). The CoNLL 2008-2009 shared tasks introduced a variant where semantic dependencies are annotated rather than phrasal arguments (Surdeanu et al., 2008; Hajič et al., 2009). Similar approaches (Das et al., 2014; Hermann et al., 2014) have been applied to frame-semantic parsing using FrameNet conventions (Baker et al., 1998). We treat PropBank and FrameNet annotations in a common framework, similar to Hermann et al. (2014).

Most prior work on SRL rely on syntactic parses provided as input and use locally estimated classifiers for each span-role pair that are only combined at prediction time.[2] This is done by picking the highest scoring role for each span, subject to a set of structural constraints, such as avoiding overlapping arguments and repeated core roles. Typically, these constraints have been enforced by integer linear programming (ILP), as in Punyakanok et al. (2008). Täckström et al. (2015) interpreted this as a graphical model with local factors for each span-role pair, and global factors that encode the structural constraints. They derived a dynamic program (DP) that enforces most of the constraints proposed by Punyakanok et al. and showed how the DP can be used to take these constraints into account during learning. Here, we use an identical graphical model, but extend the model of Täckström et al. by replacing its linear potential func-

[1] We borrow the term "lexical unit" from the frame semantics literature. The CoNLL 2005 dataset is restricted to verbal lexical units, while the CoNLL 2009 and 2012 datasets contains both verbal and nominal lexical units. FrameNet has lexical units of several coarse syntactic categories.

[2] Some recent work have successfully proposed joint models for syntactic parsing and SRL instead of a pipeline approach (Lewis et al., 2015).

tions with a multi-layer neural network. A similar use of non-linear potential functions in a structured model was proposed by Do and Artières (2010) for speech recognition, and by Durrett and Klein (2015) for syntactic phrase-structure parsing.

Feature-based approaches to SRL employ hand-engineered linguistically-motivated feature templates to represent the semantic structure. Some recent work has focused on low-dimensional representations that reduce the need for intensive feature engineering and lead to better generalization in the face of data sparsity. Lei et al. (2015) employ low-rank tensor factorization to induce a compact representation of the full cross-product of atomic features; akin to this work, they represent semantic roles as real-valued vectors, but use a different scoring formulation for modeling potential arguments. Moreover, they restrict their experiments to CoNLL 2009 semantic dependencies. Roth and Woodsend (2014) improve on the state-of-the-art feature-based system of Björkelund et al. (2010) by adding distributional word representations trained on large unlabeled corpora as features.

Collobert and Weston (2007) use a neural network and do not rely on syntactic parses as input. While they use non-standard evaluation, they report accuracy similar to the ASSERT system (Pradhan et al., 2005), to which we compare in Table 4. Very recently, Zhou and Xu (2015) proposed a deep bidirectional LSTM model for SRL that does not rely on syntax trees as input; their approach achieves the best results on CoNLL 2005 and 2012 corpora to date, but unlike this work, they do not report results on FrameNet and CoNLL 2009 dependencies and do not investigate joint learning approaches involving multiple annotation conventions.

For FrameNet-style SRL, Kshirsagar et al. (2015) recently proposed the use of PropBank-based features, but their system performance falls short of the state of the art. Roth and Lapata (2015) proposed another approach exploring linguistically motivated features tuned towards the FrameNet lexicon, but their performance metrics are significantly worse than our best results.

The inspiration behind our approach stems from recent work on bilinear models (Mnih and Hinton, 2007). There have been several recent studies representing input observations and output labels with distributed representations, for example, in the WSABIE model for image annotation (Weston et al., 2011), in models for embedding labels in structured graphical models (Srikumar and Manning, 2014), and in techniques to learn joint embeddings of predicate words and their semantic frames in a vector space (Hermann et al., 2014).

3 Model

Our model for SRL performs inference separately for each marked predicate in a sentence. It assumes that the predicate has been automatically disambiguated to a semantic frame drawn from a frame lexicon, and the semantic roles of the frame are used for labeling the candidate arguments in the sentence. Formally, we are given a sentence x in which a predicate t, with lexical unit ℓ, has been marked. Assuming that the semantic frame f of the predicate has already been identified (see §4.2 for this step), we seek to predict the set of spans that correspond to its overt semantic arguments and to label each argument with its semantic role. Specifically, we model the problem as that of assigning each span $s \in \mathcal{S}$, from an over-generated set of candidate argument spans $\mathcal{S}$, to a semantic role $r \in \mathcal{R}$. The set of semantic roles $\mathcal{R}$ includes the special null role $\varnothing$, which is used to represent non-overt arguments. Thus, our algorithm performs the SRL task in one step for a single predicate frame. For the span-based SRL task, in a sentence of n words, there could be $O(n^2)$ potential arguments. For statistical and computational reasons we prune the set of spans $\mathcal{S}$ using a set of syntactically-informed heuristics from prior work (see §4.2).

3.1 Graphical Model

We make use of a graphical model that represents global assignment of arguments to their semantic roles, subject to linguistic constraints (Punyakanok et al., 2008; Täckström et al., 2015). Under this graphical model, we assume a parameterized potential function that assigns a real-valued compatibility score $g(s, r; \boldsymbol{\theta})$ to each span-role pair $(s, r) \in \mathcal{S} \times \mathcal{R}$, where $\boldsymbol{\theta}$ denotes the model parameters. Below, we consider two types of potential functions. As a baseline, similar to most prior work, one could use a simple linear function of discrete input features $g_{\text{L}}(s, r; \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \cdot \boldsymbol{\phi}(r, s, x, t, \ell, f)$, where $\boldsymbol{\phi}(\cdot)$ denotes a feature function. In this work, we instead propose a multi-layer feed-forward neural network potential function, specified in §3.2. Given these local factors, we employ the dynamic program of Täckström et al. to enforce global constraints on the inferred output.

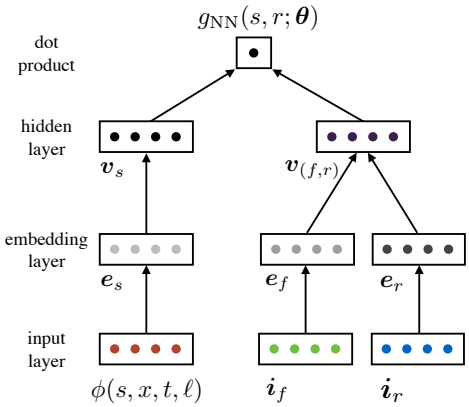


Figure 2: Neural network architecture.

• first word of s	• tag of the first word of s
• last word of s	• tag of the last word of s
• head word of s	• tag of the head word of s
• bag of words in s	• bag of tags in s
• cluster of s's head	• linear *distance* of s from t
• t's children words	• word cluster of s's head
• dependency path between s's head and t	
• subcategorization frame of s	
• *position* of s w.r.t. t (*before, after, overlap or same*)	
• predicate use voice (*active*, *passive*, or *unknown*)	
• whether the subject of t is missing (*missingsubj*)	
• *position* of s w.r.t. t (*before, after, overlap or same*)	
• word, tag, dependency label and cluster of the words immediately to the left and right of s	

Table 1: Span features $\boldsymbol{\phi}(s, x, t, \ell)$ in Figure 2.

Let $\mathcal{R}^{|\mathcal{S}|}$ denote the set of all possible assignments of semantic roles to argument spans (s_i, r_i) for $s_i \in \mathcal{S}$ that satisfy the constraints. Given a potential function $g(s, r) \triangleq g(s, r; \boldsymbol{\theta})$, the probability of a joint assignment $\boldsymbol{r} \in \mathcal{R}^{|\mathcal{S}|}$, subject to the constraints, is given by

$$p(\boldsymbol{r} \mid x, t, \ell, f) = \exp\left(\sum_{s_i \in \mathcal{S}} g(s_i, r_i) - A(\mathcal{S})\right), \quad (1)$$

where the log-partition function $A(\mathcal{S})$ sums over all satisfying joint role assignments:

$$A(\mathcal{S}) = \log \sum_{\boldsymbol{r}' \in \mathcal{R}^{|\mathcal{S}|}} \exp\left(\sum_{s_i \in \mathcal{S}} g(s_i, r_i')\right). \quad (2)$$

3.2 Neural Network Potentials

Our approach replaces the standard linear potential function $g_{\text{L}}(s, r; \boldsymbol{\theta})$ with the real-valued output of a feed forward neural network with non-linear hidden units. The network structure is outlined in Figure 2. The frame f and role r are initially encoded using a one-hot encoding as $\boldsymbol{i}_f$ and $\boldsymbol{i}_r$. In other words, $\boldsymbol{i}_f$ and $\boldsymbol{i}_r$ have all zeros except for one position at f and r respectively. These are passed through fully connected linear layers to give $\boldsymbol{e}_f$ and $\boldsymbol{e}_r$. We call these linear layers the *embedding* layers since $\boldsymbol{i}_f$ selects the embedding of the frame f and $\boldsymbol{i}_r$ for r. Next, $\boldsymbol{e}_f$ and $\boldsymbol{e}_r$ are passed through a fully connected rectified linear layer (Nair and Hinton, 2010), to obtain the final frame-role representation $\boldsymbol{v}_{(f,r)}$. For the candidate span, the process is similar. Atomic features $\boldsymbol{\phi}(s, x, t, \ell)$ for the argument span s are extracted first. (These features are the non-conjoined features used in the linear model of Täckström et al.; see Table 1 for the list). These are next passed through a fully-connected linear embedding layer to get the span embedding $\boldsymbol{e}_s$, which is subsequently passed through a fully connected rectified linear layer to obtain $\boldsymbol{v}_s$, the final span representation. The final output is the dot product of $\boldsymbol{v}_s$ and $\boldsymbol{v}_{(f,r)}$:

$$g_{\text{NN}}(s, r; \boldsymbol{\theta}) = \boldsymbol{v}_s^\top \cdot \boldsymbol{v}_{(f,r)}. \quad (3)$$

The weights of all the layers constitute the parameters $\boldsymbol{\theta}$ of the neural network. We initialize $\boldsymbol{\theta}$ randomly, with the exception of embedding parameters corresponding to words, which are initialized from pre-trained word embeddings (see §4.4 for details). We train the network as described in §3.3.[3]

Note that unlike typical linear models, the atomic span features are not explicitly conjoined with each other, the frame or the role. Instead the hidden layers learn to emulate span feature conjunctions and frame and role feature conjunctions in parallel.[4] Moreover, note that span $\boldsymbol{v}_s$ and frame-role $\boldsymbol{v}_{(f,r)}$ representations are decoupled in this model. This decoupling is important as it allows us to train a single model in a multitask setting. We demonstrate this by successfully combining PropBank and FrameNet training data, as described in §5.

3.3 Parameter Estimation

We consider two methods for parameter estimation.

[3] Various other network structures are worth investigating, such as concatenating the span, frame and role representations and passing them through fully connected layers. This treatment, for example, has been used by Chen and Manning (2014) for syntactic parsing. We leave these explorations to future work.

[4] We found that adding feature conjunctions to the network's input layer did not improve performance in practice.

Local Estimation In local estimation, we treat each span-role assignment pair $(s, r) \in \mathcal{S} \times \mathcal{R}$ as an individual binary decision problem and maximize the corresponding log-likelihood of the training set.[5] Denote by $z_{s,r} \in \{0, 1\}$ the decision variable, such that $z_{s,r} = 1$ iff span s is assigned role r. By passing the potential $g_{\text{NN}}(s, r; \boldsymbol{\theta})$ through the logistic function, we obtain the log-likelihood $l(z_{s,r}; \boldsymbol{\theta}) \triangleq \log p(z_{s,r} \mid x, t, \ell, f)$ of an individual training example. Here,

$$p(z_{s,r} \mid x, t, \ell, f) = \begin{cases} \frac{1}{1+e^{-g_{\text{NN}}(s,r;\boldsymbol{\theta})}} & \text{if } z_{s,r} = 1 \\ \frac{e^{-g_{\text{NN}}(s,r;\boldsymbol{\theta})}}{1+e^{-g_{\text{NN}}(s,r;\boldsymbol{\theta})}} & \text{if } z_{s,r} = 0 \end{cases}$$

Thus, the gold role for a given span according to the training data serves as the positive example, while all the other potential roles serve as negatives. To maximize the log-likelihood, we use Adagrad (Duchi et al., 2011). This requires the gradient of the log-likelihood with respect to the parameters $\boldsymbol{\theta}$, which can be derived using the chain rule.

Structured Estimation In structured estimation, we instead learn a globally normalized probabilistic model that takes the structural constraints into account during training. This method is closely related to the linear approach of Täckström et al. (2015), as well as to the fine-tuning of a neural CRF described by Do and Artières (2010).

We train the model by maximizing the log-likelihood of the training data, again using Adagrad. From Equation (1), we have that the log-likelihood $l(\boldsymbol{r}; \boldsymbol{\theta}) \triangleq \log p(\boldsymbol{r} \mid x, t, \ell, f)$ of a single (structured) training example $(\boldsymbol{r}, \mathcal{S}, x)$ is given by

$$l(\boldsymbol{r}; \boldsymbol{\theta}) = \sum_{s_i \in \mathcal{S}} g(s_i, r_i) - A(\mathcal{S})\,. \qquad (4)$$

By application of the chain rule, the gradient of the log-likelihood factorizes as

$$\frac{\partial l(\boldsymbol{r}; \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \frac{\partial l(\boldsymbol{r}; \boldsymbol{\theta})}{\partial g_{\text{NN}}} \frac{\partial g_{\text{NN}}}{\partial \boldsymbol{\theta}}\,, \qquad (5)$$

where we have used the shorthand g_{NN} for brevity. It is easy to show that the first term $\partial l(\boldsymbol{r}; \boldsymbol{\theta}) / \partial g_{\text{NN}}$ factors into the marginals over edges in the DP lattice, which can be computed with the forward-backward algorithm (recall that the potentials are in simple correspondence with the edge scores in the DP lattice, see Täckström et al. (2015, §4) for details). Again, the chain rule can be used to compute the gradient $\partial g_{\text{NN}} / \partial \boldsymbol{\theta}$ with respect to the parameters of each layer in the network.

[5] An alternate way to locally train the neural network would be to treat the scores as potentials in a multiclass logistic regression model and optimize log-likelihood, as is done with the locally-trained linear model from Täckström et al. (2015), but we did not investigate this alternative in this work.

3.4 Product of Experts

As we will observe in Tables 2 to 5, random initialization of the neural network parameters $\boldsymbol{\theta}$ causes variance in the performance over different runs. We found that using a straightforward product-of-experts (PoE) model (Hinton, 2002) at inference time reduces this variance and results in significantly higher performance. This PoE model is a very simple ensemble, being the factor-wise sum of the potential functions from K independently trained neural networks:

$$g_{\text{PoE}}(s, r; \boldsymbol{\theta}) = \sum_{j=1}^{K} g_{\text{NN}}^{(j)}(s, r, \boldsymbol{\theta})\,. \qquad (6)$$

where $g_{\text{NN}}^{(j)}(s, r, \boldsymbol{\theta})$ is the score from model j.

4 Experimental Setup

In this section we describe the datasets used, the required preprocessing steps, the baselines compared to and the details of our experimental setup.

4.1 Datasets and Significance Testing

We evaluate our approach on four standard datasets. For span-based SRL using PropBank conventions (Palmer et al., 2005), we evaluate on both the CoNLL 2005 shared task dataset (Carreras and Màrquez, 2005), and the larger CoNLL 2012 dataset derived from the OntoNotes 5.0 corpus (Weischedel et al., 2011). We also evaluate our model on the CoNLL 2009 shared task dataset (Hajič et al., 2009), that annotates roles for semantic dependencies, rather than full argument spans. For the CoNLL datasets, we use the standard training, development and test sets. For frame-semantic parsing using FrameNet conventions (Baker et al., 1998), we follow Das et al. (2014) and Hermann et al. (2014) in using the full-text annotations of the FrameNet 1.5 release and follow their data splits.

We use the standard evaluation scripts for each task and use a paired bootstrap test (Efron and Tibshirani, 1994) to assess the statistical significance of the results. For brevity, we only give the p-values for the observed differences between our best and second best models on each of the test sets.

4.2 Preprocessing and Frame Identification

All datasets are preprocessed with a part-of-speech tagger and a syntactic dependency parser, both trained on the CoNLL 2012 training split, after converting the constituency trees to Stanford-style dependencies (De Marneffe and Manning, 2013). The tagger is based on a second-order conditional random field (Lafferty et al., 2001) with standard emission and transition features; for parsing, we use a graph-based parser with structural diversity and cube-pruning (Zhang and McDonald, 2014).

On the WSJ development set (section 22), the labeled attachment score of the parser is 90.9% while the part-of-speech tagger achieves an accuracy of 97.2%. On the CoNLL 2012 development set, the corresponding scores are 90.2% and 97.3%. Both the tagger and the parser, as well as the SRL models use word cluster features (see Table 1). Specifically, we use the clusters with 1000 classes from Turian et al. (2010), which are induced with the Brown algorithm (Brown et al., 1992). To generate the candidate arguments $\mathcal{S}$ (see §3.2) for the CoNLL 2005 and 2012 span-based datasets, we follow Täckström et al. (2015) and adapt the algorithm of Xue and Palmer (2004) to use dependency syntax. For the dependency-based CoNLL 2009 experiments, we modify our approach to assume single length spans and treat every word of the sentence as a candidate argument. For FrameNet, we follow the heuristic of Hermann et al. (2014).

As mentioned in §3, we automatically disambiguate the predicate frames. For FrameNet, we use an embedding-based model described by Hermann et al. (2014). For PropBank, we use a multiclass log-linear model, since Hermann et al. did not observe better results with the embedding model.

To ensure a fair comparison with the closest linear model baseline, we ensured that the preprocessing steps, the argument candidate generation algorithm for the span-based datasets and the frame identification methods are identical to Täckström et al. (2015, §3.2, §6.2-§6.3).

4.3 Baseline Systems

In addition to comparing to Täckström et al. (2015), whose setup is closest to ours, we also compare to prior state-of-the-art systems from the literature.

For CoNLL 2005, we compare to the best non-ensemble and ensemble systems of Surdeanu et al. (2007), Punyakanok et al. (2008) and Toutanova et al. (2008). The ensemble variants of these systems use multiple parses and multiple SRL systems to leverage diversity. In contrast to these ensemble systems, our product-of-experts model uses only a single architecture and one syntactic parse; the constituent models differ only in random initialization. We also compare with the recent deep bidirectional LSTM model of Zhou and Xu (2015).

For CoNLL 2012, we compare to Pradhan et al. (2013), who report results with the (non-ensemble) ASSERT system (Pradhan et al., 2005), and to the model of Zhou and Xu (2015).

For CoNLL 2009, we compare to the top system from the shared task (Zhao et al., 2009), two state-of-the-art systems that employ a reranker (Björkelund et al., 2010; Roth and Woodsend, 2014), and the recent tensor-based model of Lei et al. (2015). We also trained the linear model of Täckström et al. on this dataset (their work omitted this experiment), as a baseline.

Finally, for the FrameNet experiments, we compare to the state-of-the-art system of Hermann et al. (2014), which combines a frame-identification model based on WSABIE (Weston et al., 2011) with a log-linear role labeling model.

4.4 Hyperparameters and Initialization

There are several hyperparameters in our model (§3.2). First, the span embedding dimension of $\boldsymbol{e}_s$ was fixed to 300 to match the dimension of the pre-trained GloVe word embeddings from Pennington et al. (2014) that we use to initialize the embeddings of the word-based features in $\boldsymbol{\phi}(s, x, t, \ell)$. Preliminary experiments showed random initialization of the word-based embeddings to be inferior to pre-trained embeddings. The remaining model parameters were randomly initialized. The frame embedding dimension was chosen from $\{100, 200, 300, 500\}$, while the hidden layer dimension was chosen from $\{300, 500\}$. For PropBank, we fixed the role embedding dimension to 27, which is the number of semantic roles in PropBank datasets (ignoring the AA role, that appears with negligible frequency). For FrameNet, the role embedding dimension was chosen from $\{100, 200, 300, 500\}$. In the Adagrad algorithm, the mini-batch size was fixed to 100 for local estimation (§3.3). For structured estimation (§3.3), a batch size of one was used, since each structured instance contains multiple local factors. The learning rate was chosen from $\{0.1, 0.2, 0.5, 1.0\}$ for local learning and from $\{0.01, 0.02, 0.05, 0.1\}$ for struc-

	WSJ Dev				WSJ Test				Brown Test			
Method	P	R	F1	Comp.	P	R	F1	Comp.	P	R	F1	Comp.
Surdeanu (Single)	–	–	–	–	79.7	74.9	77.2	52.0	–	–	–	–
Surdeanu (Ensemble)	–	–	–	–	87.5	74.7	80.6	51.7	81.8	61.3	70.1	34.3
Toutanova (Single)	–	–	77.9	**57.2**	–	–	79.7	**58.7**	–	–	67.8	39.4
Toutanova (Ensemble)	–	–	78.6	58.7	81.9	78.8	80.3	60.1	–	–	68.8	40.8
Punyakanok (Single)	–	–	–	–	77.1	75.5	76.3	–	–	–	–	–
Punyakanok (Ensemble)	80.1	74.8	77.4	50.7	82.3	76.8	79.4	53.8	73.4	62.9	67.8	32.3
Täckström (Local)	81.3	74.8	77.9	52.4	82.6	76.4	79.3	54.3	74.0	66.8	70.2	38.4
Täckström (Struct.)	81.2	76.2	78.6	54.4	82.3	77.6	79.9	56.0	74.3	68.6	71.3	39.8
Zhou	79.7	**79.4**	**79.6**	–	**82.9**	**82.8**	**82.8**	–	70.7	68.2	69.4	–
This work (Local)	81.4	75.6	78.4	53.7	82.3±0.4	76.8±0.5	79.4±0.1	55.1±0.6	74.1±0.6	68.0±0.7	70.9±0.3	39.1±0.8
This work (Struct.)	80.7	76.1	78.3	54.1	81.8±0.5	77.3±0.3	79.4±0.2	55.6±0.5	73.8±0.7	68.8±0.6	71.2±0.3	40.5±0.8
This work (Local, PoE)	**82.0**	76.6	79.2	55.2	**82.9**	77.8	80.3*	56.7	**75.2**	69.1	72.0	40.8
This work (Struct., PoE)	81.2	76.7	78.9	55.1	82.5	78.2	80.3*	57.3*	74.5	**70.0**	**72.2****	**41.3**

Table 2: PropBank-style SRL results on CoNLL 2005 data. Bold font indicates the best system using a single or no syntactic parse, while the best scores among all systems are underlined. Results from prior work are taken from the respective papers, and '–' indicates performance metrics missing in the original publication. Statistical significance was assessed for F1 and Comp. on the WSJ and Brown test sets with $p < 0.01$ ($*$) and $p < 0.05$ ($**$).

	Excluding predicate senses			Including predicate senses	
	WSJ Dev	WSJ Test	Brown Test	WSJ Test	Brown Test
CoNLL-2009 1st place	–	82.1	69.8	86.2	74.6
Björkelund et al., 2010 + reranking	80.5	82.9	70.9	86.9	75.7
Roth and Woodsend, 2014 + reranking	–	82.1	71.1	86.3	**75.9**
Lei et al. 2015	81.0	82.5	70.8	86.6	75.6
Täckström et al. 2015 (Local)	81.4	83.0	71.2	86.9	74.8
Täckström et al. 2015 (Struct.)	82.4	83.7	72.3	87.3	75.5
This work (Local)	81.2±0.2	82.7±0.3	71.9±0.4	86.7±0.2	75.2±0.3
This work (Struct)	82.3±0.1	83.6±0.1	71.9±0.3	87.3±0.1	75.2±0.2
This work (Local, PoE)	82.4	83.8	**72.8**	87.5	**75.9**
This work (Struct., PoE)	**83.0***	**84.3***	72.4	**87.8***	75.5

Table 3: PropBank-style *semantic dependency* SRL results (labeled F1) on the CoNLL 2009 data set. Bold font indicates the best system. Statistical significance was assessed with $p < 0.01$ ($*$).

tured learning.[6] All hyperparameters were tuned on the respective development sets for each dataset with a straightforward grid-search procedure. In the product-of-experts setup, we train $K = 10$ models, each with a different random seed, and combine them at inference time (see Equation (6)).

5 Empirical Results

Table 2 shows results on the CoNLL 2005 development set and the WSJ and Brown test sets. Our individual neural network models are on par with the best linear single-system baselines that use carefully chosen feature combinations, but has variance across reruns. On the WSJ test set, the product-of-experts model featuring neural networks trained with structured learning achieves higher F_1-score than all non-ensemble baselines, except the LSTM model of Zhou and Xu. It is on par and at times better than ensemble baselines that use diverse syntactic parses. The PoE model outperforms all baselines on the Brown test set, exhibiting its generalization power on out-of-domain text. Overall, using structured learning improves recall at a slight expense of precision when compared to local learning, leading to an increase in the complete argument structure accuracy (*Comp.* in the tables).

Table 3 shows results on the CoNLL 2009 task. Following Lei et al. (2015), we present results using the official evaluation script, along with additional metrics that do not count frame predictions. Note that the linear baseline of Täckström et al.

[6] We observed a strong interaction between learning rate and mini-batch size. Since the number of factors per frame structure is much larger than 100, lower learning rates are better suited for structured estimation.

Method	CoNLL 2012 Development P	R	F1	Comp.
Täckström (Local)	80.6	77.1	78.8	59.0
Täckström (Struct.)	80.5	77.8	79.1	60.1
Zhou	–	–	**81.1**	–
This work (Local)	80.4	77.3	78.8	59.0
This work (Struct)	80.6	77.8	79.2	59.8
This work (Local, PoE)	**81.0**	78.3	79.6	60.6
This work (Struct., PoE)	**81.0**	**78.5**	79.7	**60.9**
Method	**CoNLL 2012 Test P**	**R**	**F1**	**Comp.**
Pradhan	**81.3**	70.5	75.5	51.7
Pradhan, revised	78.5	76.6	77.5	55.8
Täckström (Local)	80.9	77.7	79.2	60.9
Täckström (Struct.)	80.6	78.2	79.4	61.8
Zhou	–	–	**81.3**	–
This work (Local)	80.6±0.3	77.8±0.2	79.2±0.1	60.8±0.3
This work (Struct.)	80.9±0.2	78.4±0.2	79.6±0.1	61.7±0.2
This work (Local, PoE)	**81.3**	78.8	80.0	62.4
This work (Struct., PoE)	81.2	**79.0**	80.1*	**62.6***

Table 4: PropBank-style SRL results on the CoNLL 2012 development and test sets. Results from prior work are taken from the respective papers, and '–' indicates performance metrics missing in the original publication. Significance was assessed for F1 and Comp. on the test set with $p < 0.01$ ($*$).

Method	FrameNet Development P	R	F1	Comp.
Hermann	78.3	64.5	70.8	–
Täckström (Local)	**80.7**	62.9	70.7	31.2
Täckström (Struct.)	79.6	64.1	71.0	33.3
This work (Local)	78.6	64.6	70.9	32.0
This work (Struct.)	79.6	63.9	70.9	31.8
This work (Local, PoE)	79.0	65.0	71.3	33.1
This work (Struct., PoE)	79.0	64.4	71.0	32.3
This work (Local, PoE, Joint)	79.4	**65.8**	**72.0**	**34.5**
This work (Struct., PoE, Joint)	78.8	65.4	71.5	33.5
Method	**FrameNet Test P**	**R**	**F1**	**Comp.**
Hermann	74.3	66.0	69.9	–
Täckström (Local)	**76.1**	64.9	70.1	33.0
Täckström (Struct.)	75.4	65.8	70.3	33.8
This work (Local)	73.9±0.6	66.4±0.4	69.9±0.3	33.4±0.6
This work (Struct.)	74.8±0.2	65.5±0.2	69.9±0.1	33.0±0.3
This work (Local, PoE)	74.3	66.9	70.4	33.9
This work (Struct., PoE)	74.6	66.3	70.2	33.3
This work (Local, PoE, Joint)	75.0	**67.3**	**70.9****	**35.4***
This work (Struct., PoE, Joint)	74.2	67.2	70.5	34.2

Table 5: Joint frame and argument identification results for FrameNet. Statistical significance was assessed for F1 and Comp. on the test set with $p < 0.01$ ($*$) and $p < 0.05$ ($**$).

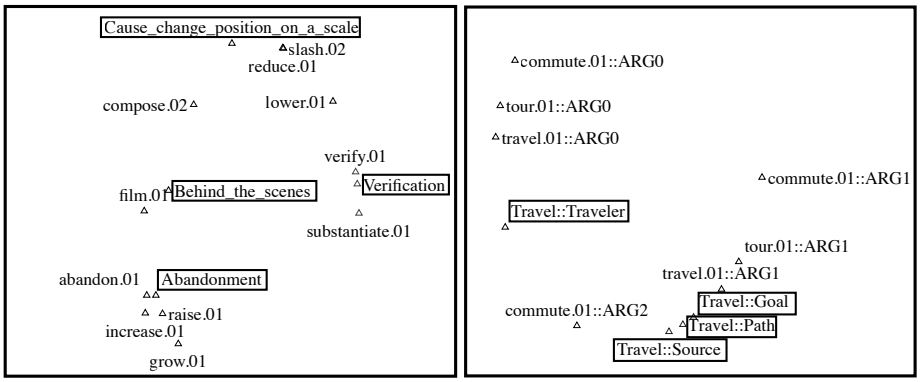
outperforms most prior work, including ones that employs rerankers, except on the Brown test set. Our neural network model performs even better, achieving state-of-the-art results on all metrics.

Table 4 shows the results on the span-based CoNLL 2012 data. The trends observed on the CoNLL 2005 data hold here as well, with structured training yielding an increase in precision at the cost of a small drop in recall. This leads to improvements in both F_1 score and complete structure accuracy. The product-of-experts model trained with structured learning here yields results better than the ASSERT system (Pradhan et al., 2013), but akin to CoNLL 2005, our system falls short in comparison to Zhou and Xu's F_1-score. In contrast to the smaller CoNLL 2005 data, even our single (non-PoE) model outperforms the linear model of Täckström et al. (2015) on the CoNLL 2012 data. We hypothesize that the relative abundance of the latter counteracts the risk for overfitting of the larger number of parameters in our model.

Finally, Table 5 shows the results on FrameNet data, which is very small in size. Here, structured learning does not help and in fact leads to a small drop in performance. Our locally-trained neural network model performs comparably to the linear model of Täckström et al. (2015). However we achieve significant improvements in both F_1-score and full structure accuracy by training our model with a dataset composed of both FrameNet and CoNLL 2005 data.[7] The ability to train in this multitask setting is a unique capability of our approach, and yields state-of-the-art results for FrameNet.

Figure 4 shows the effect of adding increasing amount of CoNLL 2005 data to supplement the FrameNet training corpus in this multitask setting. The Y-axis plots F_1-score on the development data averaged across runs for the local non-PoE model. With increasing amount of PropBank data, performance increases in small steps, and peaks when all the data is added. This shows that with more PropBank data we could further improve performance on the FrameNet task; we leave further exploration of multitask learning of predicate argument structures, including multilingual settings, to future work.

[7] The joint model does not improve results for PropBank. This is likely due to the much larger CoNLL 2005 training set, compared to the FrameNet data (39,832 training sentences in the former as opposed to 3,256 sentences in the latter).



(a) (b)

Figure 3: Two-dimensional t-SNE projections (Van der Maaten and Hinton, 2008) of joint PropBank and FrameNet (boxed) embeddings of frames (a) and frame-role pairs (b).

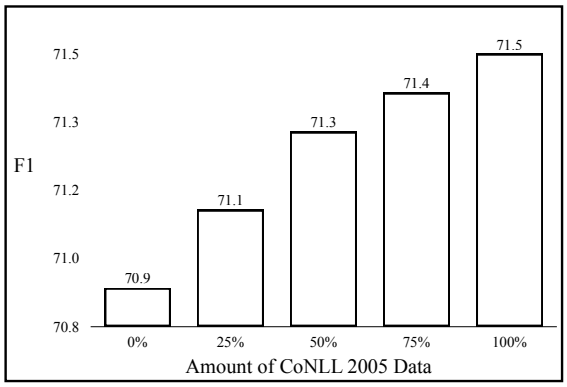


Figure 4: F_1 score on the FrameNet development data averaged over runs versus the percentage of CoNLL 2005 data used to append the FrameNet training corpus. For this plot, we used the locally trained non-PoE model.

5.1 Qualitative Analysis of Embeddings

Figure 3 shows example embeddings from the model trained jointly on FrameNet and PropBank annotations. Figure 3a shows the proximity of the learned embeddings $\boldsymbol{e}_f$ of frames from both FrameNet and PropBank. Figure 3b shows the embeddings for frame-role pairs $\boldsymbol{v}_{(f,r)}$ (the output of the hidden rectified linear layer). Here, we fix the FrameNet frame Travel and the similar PropBank frames commute.01, tour.01 and travel.01 are visualized along with their semantic roles. We observe that the model learns very similar embeddings for the semantically related roles across both datasets. Note that there is a clear separation of the agentive roles from the others for both conventions and how the FrameNet and PropBank counterparts of each type of role are proximate in vector space.

6 Conclusion

We presented a neural network model for semantic role labeling that learns to embed both inputs and outputs in the same vector space. We considered both local and structured training methods for the network parameters from supervised SRL data. Empirically, our approach achieves state-of-the-art results on two standard datasets with a product of experts model, while approaching the performance of a recent deep recurrent neural network model on two other datasets. By training the model jointly on both FrameNet and PropBank data, we achieve the best result to date on the FrameNet test set. Finally, qualitative analysis indicates that the model represents semantically similar annotations with proximate vector-space embeddings.

Acknowledgments

We thank Tom Kwiatkowski, Slav Petrov and Fernando Pereira for comments on early drafts. We are also thankful to Mike Lewis, Mark Yatskar and Luke Zettlemoyer for valuable feedback. Finally, we thank the three anonymous reviewers for suggestions that enriched the final version of the paper.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of ACL*.

Anders Björkelund, Bernd Bohnet, Love Hafdell, and Pierre Nugues. 2010. A high-performance syntactic and semantic dependency parser. In *Proceedings of ICCL: Demonstrations*.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18(4):467–479.

Xavier Carreras and Lluís Màrquez. 2004. Introduction to the CoNLL-2004 shared task: Semantic role labeling. In *Proceedings of CoNLL*.

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

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of EMNLP*.

Ronan Collobert and Jason Weston. 2007. Fast semantic extraction using a novel neural network architecture. In *Proceedings of ACL*.

Dipanjan Das, Desai Chen, Andre F. T. Martins, Nathan Schneider, and Noah A. Smith. 2014. Frame-semantic parsing. *Computational Linguistics*, 40(1):9–56.

Marie-Catherine De Marneffe and Christopher D Manning, 2013. *Stanford typed dependencies manual*.

Trinh-Minh-Tri Do and Thierry Artières. 2010. Neural conditional random fields. In *Proceedings of AISTATS*.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Greg Durrett and Dan Klein. 2015. Neural CRF parsing. In *Proceedings of ACL*.

Bradley Efron and Robert J Tibshirani. 1994. *An introduction to the bootstrap*. CRC press.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, et al. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of CoNLL*.

Karl Moritz Hermann, Dipanjan Das, Jason Weston, and Kuzman Ganchev. 2014. Semantic frame identification with distributed word representations. In *Proceedings of ACL*.

Geoffrey E. Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

Meghana Kshirsagar, Sam Thomson, Nathan Schneider, Jaime Carbonell, Noah A. Smith, and Chris Dyer. 2015. Frame-semantic role labeling with heterogeneous annotations. In *Proceedings of ACL*.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of ICML*.

Tao Lei, Yuan Zhang, Lluís Márquez, Alessandro Moschitti, and Regina Barzilay. 2015. High-order low-rank tensors for semantic role labeling. In *Proceedings of NAACL*.

Mike Lewis, Luheng He, and Luke Zettlemoyer. 2015. Joint A* CCG parsing and semantic role labelling. In *Proceedings of EMNLP*.

Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of ICML*.

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted Boltzmann machines. In *Proceedings of ICML*.

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

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of EMNLP*.

Sameer Pradhan, Kadri Hacioglu, Valerie Krugler, Wayne Ward, James H Martin, and Daniel Jurafsky. 2005. Support vector learning for semantic argument classification. *Machine Learning*, 60(1-3):11–39.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of CoNLL*.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2):257–287.

Michael Roth and Mirella Lapata. 2015. Efficient inference and structured learning for semantic role labeling. *Transactions of the Association for Computational Linguistics*, 3:449–460.

Michael Roth and Kristian Woodsend. 2014. Composition of word representations improves semantic role labelling. In *Proceedings of EMNLP*.

Vivek Srikumar and Christopher D Manning. 2014. Learning distributed representations for structured output prediction. In *Proceedings of NIPS*.

Mihai Surdeanu, Lluís Màrquez, Xavier Carreras, and Pere Comas. 2007. Combination strategies for semantic role labeling. *Journal of Artificial Intelligence Research*, 29:105–151.

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 CoNLL*.

Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Efficient inference and structured learning for semantic role labeling. *Transactions of the Association for Computational Linguistics*, 3:29–41.

Kristina Toutanova, Aria Haghighi, and Christopher D Manning. 2008. A global joint model for semantic role labeling. *Computational Linguistics*, 34(2):161–191.

Joseph Turian, Lev Ratinov, and Yoshua Bengio. 2010. Word representations: a simple and general method for semi-supervised learning. In *Proceedings of ACL*.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(85).

Ralph Weischedel, Eduard Hovy, Martha Palmer, Mitch Marcus, Robert Belvin, Sameer Pradhan, Lance Ramshaw, and Nianwen Xue. 2011. OntoNotes: A large training corpus for enhanced processing. In J. Olive, C. Christianson, and J. McCary, editors, *Handbook of Natural Language Processing and Machine Translation*. Springer.

Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. WSABIE: Scaling up to large vocabulary image annotation. In *Proceedings of IJCAI*.

Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. In *Proceedings of EMNLP*.

Hao Zhang and Ryan McDonald. 2014. Enforcing structural diversity in cube-pruned dependency parsing. In *Proceedings of ACL*.

Hai Zhao, Wenliang Chen, Chunyu Kity, and Guodong Zhou. 2009. Multilingual dependency learning: A huge feature engineering method to semantic dependency parsing. In *Proceedings of CoNLL*.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proceedings of ACL*.