A Joint Sequential and Relational Model for Frame-Semantic Parsing

Bishan Yang
Machine Learning Department
Carnegie Mellon University
bishan@cs.cmu.edu

Tom Mitchell
Machine Learning Department
Carnegie Mellon University
tom.mitchell@cs.cmu.edu

Abstract

We introduce a new method for frame-semantic parsing that significantly improves the prior state of the art. Our model leverages the advantages of a deep bidirectional LSTM network which predicts semantic role labels word by word and a relational network which predicts semantic roles for individual text expressions in relation to a predicate. The two networks are integrated into a single model via knowledge distillation, and a unified graphical model is employed to jointly decode frames and semantic roles during inference. Experiments on the standard FrameNet data show that our model significantly outperforms existing neural and non-neural approaches, achieving a 5.7 F1 gain over the current state of the art, for full frame structure extraction.

1 Introduction

One way to represent meaning is through organization of semantic structures. Consider the following sentences "John sells Marry a car." and "Mary buys a car from John.". While having different syntactic structures, they express the same type of event that involves a buyer, a seller, and goods. Such meaning can be represented using semantic frames – structured representations that characterize events, scenarios, and the participants. Researchers have developed FrameNet (Baker et al., 1998; Fillmore et al., 2003), a large lexical database of English that comes with sentences annotated with semantic frames. It has been considered a valuable resource for Natural Language Processing and useful for studying tasks such as information extraction, machine translation, and question answering (Surdeanu et al., 2003; Shen and Lapata, 2007; Liu and Gildea, 2010).

Here we consider the task of automatic extraction of semantic frames as defined in FrameNet. This include *target identification* – identifying frame-evoking predicates, *frame identification* – identifying which frame each predicate evokes, and *semantic role labeling* (SRL) – identifying phrasal arguments of each evoked frame and labeling them with the frame's semantic roles. Consider the sentence "We decided to treat the patient with combination chemotherapy.". Here "decided" evokes the DECIDING frame and "treat" evokes the CURE frame. Each frame takes a set of arguments that fill the semantic roles of the frame, as illustrated below:

> [We$_{\text{COGNIZER}}$] **decided** [to treat the patient with combination chemotherapy$_{\text{DECISION}}$].

> [We$_{\text{HEALER}}$] decided to **treat** [the patient$_{\text{PATIENT}}$] [with combination chemotherapy$_{\text{TREATMENT}}$].

We address frame identification and semantic role labeling in this work.[1] Frame identification can be addressed as a word sense disambiguation problem, while semantic role labeling can be formulated as a structured prediction problem. We train different neural network models for these two problems, and interpret their outputs as factors in a graphical model for performing joint inference over the distribution of frames and semantic roles.

Specifically, our frame identification model is a simple multi-layer neural network that learns ap-

[1] We do not consider target identification due to the lack of consistent labeled data (Das et al., 2014).

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1258–1267
Copenhagen, Denmark, September 7–11, 2017.

propriate feature representations for frame disambiguation. Our SRL model is an integrated model of an LSTM-based network that learns to predict semantic roles on a word-by-word basis and a multi-layer network that learns to directly predict semantic roles for individual text spans in relation to a given predicate. The sequential neural network is powerful for modeling sentence-level information while the relational neural network is good at capturing span-level dependencies between predicate and arguments. To leverage the power of these two networks, we transfer the knowledge in the sequential model, encoded as its predictive distributions. Specifically, we do this by training a single relational model with an objective that measures both its prediction accuracy with respect to the true semantic role labels, and its match to the probability distributions provided by the sequential model.

We evaluate our models for frame identification, SRL, and full structure extraction on the FrameNet 1.5 data. Our full model achieves 76.6 F1, a 5.7 absolute gain over the prior state of the art. We also evaluate our SRL model on CoNLL 2005. It demonstrates strong performance that is close to the best published results. Error analysis further confirms the benefits of integrating sequential and relational models and performing joint inference over frames and semantic roles.

2 Related Work

Research on automatic semantic structure extraction has been widely studied since the pioneering work of Gildea and Jurafsky (2002). This work focuses on extracting semantic frames defined in FrameNet (Baker et al., 1998), which includes predicting frame types and frame-specific semantic roles. Our model can be easily adapted to predict PropBank-style semantic roles (Palmer et al., 2005), where role labels are generic instead of frame-specific.

The core problem in semantic frame extraction is semantic role labeling (SRL). Earlier SRL systems employ linear classifiers which rely heavily on hand-engineered feature templates to represent argument structures (Johansson and Nugues, 2007; Das et al., 2010; Das, 2014). Recent work has exploited neural networks to learn better feature representations. Roth and Woodsend (2014) improves the feature-based system by adding word embeddings as features. Roth and Lapata (2016) further includes dependency path embeddings as features. FitzGerald et al. (2015) embeds the standard SRL features into a low-dimensional vector space using a feed-forward neural network and demonstrates state-of-the-art results on FrameNet.

Different neural network architectures have also been explored for SRL. Collobert et al. (2011) first applies a convolutional neural network to extract features from a window of words. Zhou and Xu (2015) employs a deep bi-directional LSTM (DB-LSTM) network and achieves state-of-the-art results on PropBank-style SRL. Recently, Swayamdipta et al. (2016) employs stack LSTMs (Dyer et al., 2015) for joint syntactic-semantic dependency parsing. He et al. (2017) recently proposed further improvements to the DB-LSTM architecture which significantly improve the state of the art results on PropBank SRL.

In order to enforce structural consistency, most existing work applies different types of structural constraints during inference. The inference problem are typically solved via Integer Linear Programming (ILP) (Punyakanok et al., 2008). Täckström et al. (2015) improves the inference efficiency with a dynamic programming algorithm that encodes tractable global constraints. Recently, Belanger et al. (2017) models SRL using end-to-end structured prediction energy networks and demonstrates benefits of accounting for complex structural dependencies during training. In this work, we explicitly encode structural constraints as factors in a graphical model, and adopt the Alternating Directions Dual Decomposition (AD^3) algorithm (Martins et al., 2011) for efficient inference.

3 Overview

We aim to extract frame-semantic structures from text. Each semantic frame contains a frame-evoking predicate, its frame type, the arguments of the predicate, and their semantic roles.

Both FrameNet (Baker et al., 1998) and PropBank (Palmer et al., 2005) provide sentences annotated with predicates and the semantic roles of arguments of the predicates, but there are some differences. In FrameNet, a semantic frame can be evoked by a set of lexical units. For example, the COMMERCE_BUY frame can be evoked by *buy.v*, *purchase.n*, and *purchase.v*. Each frame is also associated with a set of roles, some of which are core roles (necessary components) of the frame.

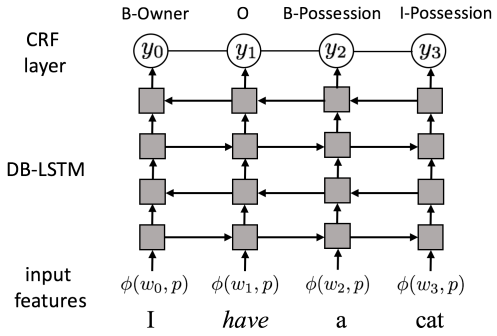

Figure 1: DB-LSTM network (four layers) with a CRF prediction layer. The network learns to predict a sequence of argument role labels given a sentence (e.g., "I have a cat") and a predicate (e.g., "have").

For example, the COMMERCE_BUY frame contains core roles such as BUYER and GOODS, and non-core roles such as MONEY and MEANS. In PropBank, a semantic frame is corresponding to a verb senses. Each verb sense is associated with a set of semantic roles. For example, the verb sense *buy.01* is associated with roles A0 (i.e., agent), A1 (i.e., patient), A2 (i.e., instrument), etc. The semantic roles in PropBank use generic labels. There are about 30 different role labels in total (vs $\sim 10^3$ role labels in FrameNet). Among them 7 are core role labels (A0-A5 and AA) and the rest are non-core (modifier) roles (e.g., the locative role LOC and the temporal role TMP).

In the rest of the paper, we first describe our models for SRL (§ 4), including a sequential neural model, a relational neural model, and the integration of the two. Then, we present our frame identification model (§ 5), followed by a joint inference algorithm for full frame-semantic structure extraction that enforces structural constraints among predicates and arguments (§ 6).

4 Semantic Role Labeling

Given a predicate and its frame, we seek to identify arguments of the predicate and their semantic roles in relation to the predicate's frame. Denote a predicate as p, its frame as f, and a sentence as x. We want to output a set of argument spans $A = \{a_1, ..., a_k\}$, where each a_i is labeled with a semantic role that takes values from a set of role labels $\mathcal{R}_f$ with respect to the frame f.

4.1 Sequential Neural Model

The SRL task can be formulated as a sequence labeling problem, where the semantic role labels are encoded using the "IOB" tagging scheme, as in (Collobert et al., 2011; Zhou and Xu, 2015), where "I" indicates the inside of a chunk, "B" indicates the beginning of a chunk, and "O" indicates being outside of a chunk.

We employ DB-LSTM, a deep bidirectional Long Short-Term Memory neural network with a Condidtional Random Field (CRF) layer introduced by Zhou and Xu (2015) for PropBank-style SRL. The architecture is illustrated in Figure 1. In this work, we adapt it to perform both FrameNet-style and PropBank-style SRL.

At each time step t, the DB-LSTM network is provided with a set of input features $\phi(w_t, p)$, including the current word w_t, the predicate word p, and a position mark that denotes whether the current word is in the neighborhood of the predicate (within a window of 5 words)[2]. Each word feature is associated with a parameter vector which is initialized using the pre-trained paraphrastic word embeddings (Wieting et al., 2015). The input representation at time step t is the concatenation of the above features. As proposed in (Zhou and Xu, 2015), we stack 8 layers of the LSTM unit to produce the hidden representation for each time step. Then, we employ a CRF layer on top to estimate the sequence-level label distributions.

During training, we minimize the negative conditional log-likelihood of N training examples. Each example consists of a sentence x, a predicate p, and a label sequence $\mathbf{y} = \{y_1, ..., y_n\}$, where n is the length of the sentence. The conditional probability is given by:

$$P_{seq}(\mathbf{y} \mid p, f; \boldsymbol{\theta}) = \frac{1}{Z_f} \exp\Big(\sum_{t=1}^{n} C_{t,y_t} + \sum_{t=0}^{n} T_{y_t,y_{t+1}}\Big) \quad (1)$$

where Z_f is a normalization constant depending on the frame f, as we only normalize over role label sequences that are compatible with the frame. For PropBank-style SRL, we simply drop the dependency on f and compute normalization over all possible role label sequences. C_{t,y_t} is the score output by DB-LSTM for assigning the t-th word

[2] We did not use the predicate context features as in Zhou and Xu (2015) since they did not improve performance in our implementation.

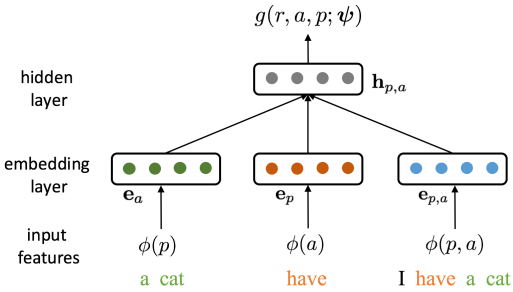

Figure 2: A relational network architecture. The network learns to predict a relation between a predicate p and an argument a given the predicate-argument pair and the sentence that contains it.

with label y_t and $T_{y_t, y_{t+1}}$ is the score of transitioning from label y_t to y_{t+1}. $\boldsymbol{\theta}$ denotes the model parameters, including the DB-LSTM parameters and the transition matrix T.

4.2 Relational Neural Model

An alternative way to formulate the SRL problem is to enumerate all possible argument spans for a given predicate and employ multi-class classification on every argument span. We describe how to obtain candidate argument spans in Section § 7.2.

Denote a set of candidate argument spans as $\tilde{\mathcal{A}}$. For each argument span $a \in \tilde{\mathcal{A}}$, we seek to estimate the conditional probability given by:

$$P_{rel}(r \mid a, p, f; \boldsymbol{\psi}) = \frac{\exp(g(r, a, p; \boldsymbol{\psi}))}{\sum_{r' \in \mathcal{R}_f \cup \varnothing} \exp(g(r', a, p; \boldsymbol{\psi}))} \quad (2)$$

where $g(r, a, p; \boldsymbol{\psi})$ is a potential function for scoring the assignment of semantic role r to an argument span a with respect to predicate p, $\boldsymbol{\psi}$ denotes the model parameters, $\mathcal{R}_f$ is a set of valid semantic roles with respect to frame f and $\varnothing$ is an empty class that indicates invalid semantic roles.

We estimate g using a neural network as depicted in Figure 2. The inputs to the network are discrete features: $\phi(a)$ denotes argument-specific features, which include words within the argument span, the dependents of the argument's head, and their dependency labels; $\phi(p)$ denotes predicate-specific features, which include the predicate word, its dependents, and their dependency labels; $\phi(p, a)$ denotes predicate-argument relation features, which include the words between p and a and the lexicalized shortest dependency path.

We then map these features into a low dimensional space. Specifically, we compute an embedding of the argument features: $\mathbf{e}_a = [\bar{\mathbf{v}}_w^a; \bar{\mathbf{v}}_d^a; \bar{\mathbf{v}}_l^a]$, where $\bar{\mathbf{v}}_w^a \in \mathbb{R}^k$ is the average of argument word embeddings, $\bar{\mathbf{v}}_d^a \in \mathbb{R}^k$ is the average embedding of the argument's dependents, and $\bar{\mathbf{v}}_l^a \in \mathbb{R}^k$ is the average embedding of the corresponding dependency labels. Similarly, the embedding of the predicate features is: $\mathbf{e}_p = [\bar{\mathbf{v}}_w^p; \bar{\mathbf{v}}_d^p; \bar{\mathbf{v}}_l^p]$, which is the concatenation of the average embeddings for the predicate words, the predicate's dependents, and their dependency labels. For the relational features, we have $\mathbf{e}_{p,a} = [\bar{\mathbf{v}}_w^{pa}; \mathbf{v}_{path}]$, where $\bar{\mathbf{v}}_w^{pa} \in \mathbb{R}^k$ is the average embedding for words between p and a, and $\mathbf{v}_{path} \in \mathbb{R}^k$ is a dependency path embedding, which is the final hidden state of an LSTM network that operates over the dependency path between p and a, with the input at each time step being the concatenation of a dependency label embedding and a word embedding.

The feature embeddings are then integrated through a non-linear hidden layer:

$$\mathbf{h}_{p,a} = \text{ReLu}(\mathbf{W}_{p,a} \cdot [\mathbf{e}_a; \mathbf{e}_p; \mathbf{e}_{p,a}]) \quad (3)$$

where $\mathbf{W}_{p,a}$ is an $m \times 8k$ matrix and $\text{ReLu}(x) = \max(0, x)$. Finally, we compute the potential function: $g(r, a, p; \boldsymbol{\psi}) = \mathbf{w}_r^T \mathbf{h}_{p,a}$, where $\mathbf{w}_r \in \mathbb{R}^m$ is a weight vector to be learned.

During training, we minimize the negative conditional log-likelihood of the training examples, with the conditional probability for each example given by Eq. 2.

4.3 An Integrated Model

Our integrated model is essentially a relational neural model that is learned using the knowledge distilled from the sequential model.

Note that the sequential model estimates probabilities for semantic role label sequences over words instead of over text spans. These learned probabilities carry important information about how the sequential model learns to generalize. We identify them as the learned knowledge of the sequential model. To make use of such knowledge in the relational model, we first transform the sequence distributions into span-based distributions. Specifically, we derive the marginal distribution for any given span $a = (w_s, ..., w_t)$, $1 \leq s \leq t < n$, and a non-empty semantic role label r as:

$$P_{seq}(r \mid a) = P_{seq}(y_s = B_r, ..., y_t = I_r, y_{t+1} \neq I_r \mid a) \quad (4)$$

Here we drop the dependency on p, and f for brevity. B_r, I_r, and O denote the beginning, the

inside, and the outside of the filler of role r respectively. The probability for an empty role is:

$$P_{seq}(r = \varnothing \mid a) = 1 - \sum_{r \in \mathcal{R}_f} P_{seq}(r \mid a) \quad (5)$$

After obtaining the span-based role distributions, we incorporate them into the training objective of a relational model $\tilde{P}_{rel}$ by adding a regularization term that minimizes the KL divergence:

$$L = -\log \tilde{P}_{rel}(r \mid a) + \beta KL(P_{seq} || \tilde{P}_{rel}) \quad (6)$$

which is equivalent to minimizing

$$-\log \tilde{P}_{rel}(r \mid a) + \beta \sum_{r} P_{seq}(r \mid a) \log \tilde{P}_{rel}(r \mid a)$$

where β is a weight parameter. We refer to $\tilde{P}_{rel}$ as the integrated model. At inference time, it computes the predictive distributions of semantic roles in the same way as a vanilla relational model.

5 Frame Identification

Our semantic role labeling model is conditioned on a predicate and its frame. We now describe how to estimate the probabilities of a frame f given a predicate p.

Denote $\mathcal{F}$ as a set of semantic frames, we learn to estimate the probability:

$$P_f(f \mid p) = \frac{\exp\left(u(f, p; \boldsymbol{\lambda})\right)}{\sum_{f' \in \mathcal{F}} \exp\left(u(f', p; \boldsymbol{\lambda})\right)} \quad (7)$$

The potential function $u(f, p; \boldsymbol{\lambda})$ is computed using a multi-layer neural network, whose architecture is similar to Figure 2. The input features are $\phi(p)$ as defined in § 4.2. The embedding layer computes $\mathbf{e}_p$ as described above, and the hidden layer computes:

$$\mathbf{h}_p = \text{ReLu}(\mathbf{W}_p \cdot \mathbf{e}_p)$$

where $\mathbf{W}_p$ is an $m \times 3k$ matrix. The potential function is then estimated as $u(f, p; \boldsymbol{\lambda}) = \mathbf{w}_f^T \mathbf{h}_p$, where $\mathbf{w}_f \in \mathbb{R}^m$ is a weight vector to be learned. Training is done by minimizing the negative conditional log-likelihood of the training examples where the conditional probability for each example is given by Eq. 7.

6 Joint Inference

Finally, we want to jointly assign frames and roles to all predicates and their arguments.

Given a set of predicates $\mathcal{P} = \{p_1, ..., p_N\}$ and a set of candidate argument spans $\tilde{\mathcal{A}} = \{a_1, ..., a_M\}$, we optimize the following objective:

$$\underset{\text{s.t. } (\mathbf{f},\mathbf{r}) \in \mathcal{Q}}{\arg\max} \sum_{j=1}^{N} P_f(f_j \mid p_j) \sum_{i=1}^{M} \tilde{P}_{rel}(r_i \mid a_i, p_j, f_j) \quad (8)$$

where $\mathbf{f}$ is a vector of frame assignments, $\mathbf{r}$ is a vector of role assignments, and $\mathcal{Q}$ is a constrained set of frame and role assignments.

We employ the standard structural constraints for SRL, including avoiding non-overlapping argument spans and repeated core roles for each frame. In addition, we introduce two constraints: one encodes the compatibility between frame types and semantic roles, for example, INSTRUMENT is not a valid role for the frame COMMERCIAL_TRANSACTION, and the other encodes type consistencies of semantic role fillers of different frames, e.g., the same named entity cannot play both a PERSON role and a VEHICLE role. We consider six common entity types (that are mutually exclusive): PERSON, LOCATION, WEAPON, VEHICLE, VALUE, and TIME.[3]

We solve the inference problem (8) using the AD^3 algorithm (Martins et al., 2011), which allows for more efficient constrained optimization than generic Integer Linear Programming solvers.

7 Experiment

7.1 Datasets

We evaluate our approach on semantic frame extraction using the FrameNet 1.5 release[4]. We use the same train/development/test split of the fully-annotated text documents as in previous work. We also include the partially-annotated exemplar sentences (i.e., each exemplar has only one annotated frame.) in FrameNet as training data.[5] We use the standard evaluation script that measures frame

[3] We simply check if the role name contains any of the entity type names like "person", "location". We plan to incorporate an automatic semantic typing model into our framework in future work.

[4] http://framenet.icsi.berkeley.edu

[5] Existing work also makes use of the exemplars, but mainly as a lexicon. We found that adding the exemplar sentences generally introduces a 3-4 F1 gain for FrameNet SRL.

structure extraction precision, recall and F1[6].

For PropBank-style SRL, we use the CoNLL2005 data set (Carreras and Màrquez, 2005) with the official scripts[7] for evaluation. It contains section 2-21 of WallStreet Journal (WSJ) data as training set, section 24 as development set and section 23 of WSJ concatenated with 3 sections from Brown corpus as the test set.

For data pre-processing, we parse all the sentences with the part-of-speech tagger and the dependency parser provided in the Stanford CoreNLP toolkit (Manning et al., 2014).

7.2 Argument candidate extraction

Existing work relied on either constituency syntax (Xue and Palmer, 2004) or dependency syntax (Täckström et al., 2015) to derive heuristic rules for extracting candidate arguments. Instead, we extract candidate arguments using a pre-trained sequential SRL model (described in § 4.1). Specifically, we extract the argument spans from the K-best semantic role label sequences output by the sequential model. We choose K from {5,10,20,50}. Increasing K will increase the recall of unlabeled arguments but lower the precision. We tune K based on the argument extraction performance of our relational model (in § 4.2) using the development data. In all our experiments, we set $K = 10$, which gives an unlabeled argument recall/precision of 89.6%/24.8% on FrameNet and 92.4%/29.4% on CoNLL2005.

7.3 Implementation details

All of our models are implemented using Theano on a single GPU. We set the embedding dimension k to 300 and the hidden dimension m to 100. We initialize the word embeddings using the pre-trained word embeddings from (Wieting et al., 2015) while randomly initializing the embeddings for out-of-vocabulary words and the embeddings for the dependency labels within $(-0.01, 0.01)$. All these embeddings are updated during the training process. We apply dropout to the embedding layer with rate 0.5, and train using Adam with default settings (Kingma and Ba, 2014). The weight parameter β in Eq. 6 is set to 1 in our experiments. All the models are trained for 50 epochs with early stopping based on development results.

[6] http://www.cs.cmu.edu/~ark/SEMAFOR/eval/

[7] http://www.lsi.upc.edu/~srlconll/srl-eval.pl

Model	All	Ambiguous
LOG-LINEAR WORDS	87.3	70.5
LOG-LINEAR EMBEDDING	86.7	70.3
WSABIE EMBEDDING	**88.4**	73.1
Ours (Frame Only)	88.2	**75.7**

Table 1: Accuracy results on frame identification, including results on *all* predicates and *ambiguous* predicates in the FrameNet lexicon.

For all our experimental results, we perform statistical significance tests using the paired bootstrap test (Efron and Tibshirani, 1994) with 1000 bootstrap samples of the evaluated examples, and use $*$ to indicate statistical significance ($p < 0.05$) of the differences between our best model and our second best model.

7.4 FrameNet Results

Frame Identification. We first evaluate our frame identification model in § 5. For baselines, we consider the prior state-of-the-art approach WSABIE EMBEDDING (Hermann et al., 2014), which learns feature representations based on word embeddings and dependency path embeddings using the WSABIE algorithm (Weston et al., 2011). We also include two strong baselines implemented in Hermann et al. (2014): LOG-LINEAR WORDS and LOG-LINEAR EMBEDDINGS, which are both log-linear models, one with standard linguistic features and one with embedding features. Table 1 shows the results.[8] We can see that our model in general gives competitive performance and it outperforms all the baselines on predicting frames for ambiguous predicates (i.e., seen with more than one possible frames in the FrameNet lexicon).

Semantic Role Labeling. Next, we evaluate our SRL models with gold-standard frames, so that we can focus on the performance for argument identification. Our SRL models include the sequential model described in § 4.1 (denoted as *Seq*); the relational model described in § 4.2 (denoted as *Rel*); and the integrated model described in § 4.3 (denoted as *Seq+Rel*).

Table 2 shows the results for argument span ex-

[8] We consider the FULL LEXICON evaluation setting and copy the results from the updated version of the paper from the author's website http://www.dipanjandas.com/pages/papers. Note that the set of ambiguous predicates we consider is different from the set used by Hermann et al. (2014). This is because we process the lexical units with the Stanford POS tagger.

Model	Prec.	Rec.	F1
SEMAFOR	65.6	53.8	59.1
SEMAFOR (best)	66.0	60.4	63.1
Ours (Seq)	63.4	**66.4**	64.9
Ours (Rel)	**71.8**	57.7	64.0
Ours (Seq+Rel)	70.2	60.2	**65.5***

Table 2: Argument only evaluation results on the FrameNet test set in comparison to the results in Kshirsagar et al. (2015).

Model	Prec.	Rec.	F1
SEMAFOR	78.4	73.1	75.7
Framat	80.3	71.7	75.8
Framat+context	80.4	73.0	76.5
Ours (Seq)	78.5	**79.9**	79.2
Ours (Rel)	**84.8**	75.5	80.0
Ours (Seq+Rel)	84.2	77.1	**80.5***

Table 3: Full structure extraction results on the FrameNet test set (with gold frames) in comparison to the results in Roth and Lapata (2015).

traction. Our baselines include SEMAFOR (Das et al., 2014)[9], a widely used frame-semantic parser for English, and SEMAFOR (BEST), an improved SEMAFOR system that is trained with heterogeneous resources (Kshirsagar et al., 2015). We can see that all of our models outperform these two systems in terms of F1, especially, our sequential model provides the best recall, our relation model provides the best precision, and our integrated model gives the best F1 score.

Table 3 shows results for full structure extraction (i.e., the accuracies of the frame-argument structure as a whole). We compare to the results reported in Roth and Lapata (2015). *Framat* is an open-source semantic role labeling tool provided by mate-tools (Björkelund et al., 2010), and *Framat+context* is an extension of *Framat* that uses additional context features. All of our models significantly outperform the baselines in F1. In particular, our integrated model achieves the best F1 score of 80.5%.

Full Semantic Structure Extraction. We now evaluate our models on full semantic frame extraction. Previous work implements the task in a two-stage pipeline: first apply a frame identification model to assign a frame to each predicate, and then apply a SRL model to assign a frame-specific role label or ∅ to each candidate argument span. We compare with previous work using four model variants: three are pipeline models that combine our frame identification model with each of our SRL models and *JointAll* is the joint model that simultaneously predicts frames and roles as described in § 6.

[9] http://www.cs.cmu.edu/~ark/SEMAFOR/

Model	Prec.	Rec.	F1
SEMAFOR	69.2	65.1	67.1
Framat	71.1	63.7	67.2
Framat+context	71.1	64.8	67.8
Hermann	74.3	66.0	69.9
Täckström (Struct.)	75.4	65.8	70.3
FitzGerald (Struct.)	74.8	65.5	69.9
FitzGerald (Struct., PoE)	74.6	66.3	70.2
FitzGerald (Local, PoE, Joint)	75.0	67.3	70.9
Ours (Seq)	69.6	70.9	70.2
Ours (Rel)	77.1	68.7	72.7
Ours (Seq+Rel)	77.3	71.2	74.1
Ours (JointAll)	**78.8**	**74.5**	**76.6***

Table 4: Full structure extraction results on the FrameNet test set in comparison to the previously published results.

Table 4 compares our models with previously published results. The first block shows results from Roth and Lapata (2015) and the second block shows results from FitzGerald et al. (2015). All these previous methods implements a pipeline of frame identification and semantic role labeling. The first block uses SEMAFOR for frame identification and the second block uses the WSABIE model from Hermann et al. (2014). For the semantic role labeling step, *Hermann* is a standard log-linear classification model used in Hermann et al. (2014); *Täckström (Struct.)* is a graphical model with global factors (Täckström et al., 2015); *FitzGerald (Struct.)* is an improved version of the graphical model with non-linear potential functions instead of linear ones; *FitzGerald (Struct., PoE)* further employs an ensemble with the product-of-experts (PoE) (Hinton, 2002); and *FitzGerald (Local, PoE, Joint)* indicates the best reported results in FitzGerald et al. (2015) which uses local factors and additional training data from CoNLL 2005. We can see that our sequential model alone is already close to the state of the art. Our relational model demonstrates superior performance on precision, which confirms the benefit of modeling predicate-argument interactions at the span level. The integrated model further improves over the relational model in both precision and recall. Finally, by

Method	Dev	WSJ	Brown
Surdeanu (Ensemble)	-	80.6	70.1
Toutanova (Ensemble)	78.6	80.3	68.8
Punyakanok (Ensemble)	77.4	79.4	67.8
Zhou (DB-LSTM)	79.6	**82.8**	69.4
Täckström (Struct.)	78.6	79.9	71.3
FitzGerald (Struct.)	78.3	79.4	71.2
FitzGerald (Struct., PoE)	78.9	80.3	**72.2**
Ours (Seq)	78.5	80.5	70.8
Ours (Rel)	79.2	81.4	71.3
Ours (Seq+Rel)	**80.3**	81.9	72.0*

Table 5: Semantic role labeling results on CoNLL 2005.

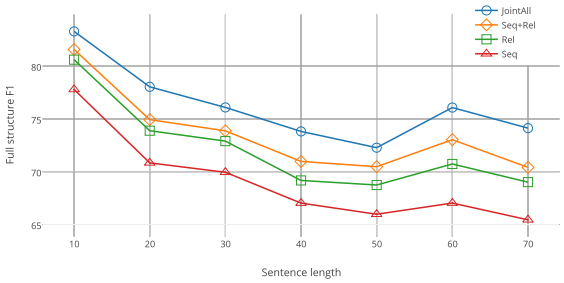

Figure 3: Full structure F1 on the FrameNet test set by the sentence length.

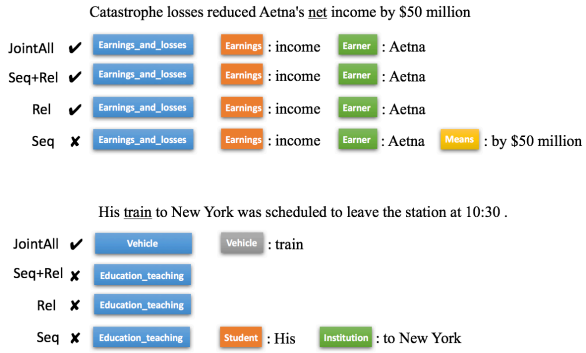

Figure 4: Examples of semantic frames output by different models.

joint inference of both frames and semantic roles, our model performs even better, achieving a 5.7 absolute F1 gain over the prior state of the art.

7.5 CoNLL Results

Table 5 shows the results of our SRL models on the CoNLL 2005 data. Our baselines include the best feature-based systems of Surdeanu et al. (2007), Toutanova et al. (2008), and Punyakanok et al. (2008), the recurrent neural network model (DB-LSTM) (Zhou and Xu, 2015), the graphical model with global factors (Täckström et al., 2015) and the improved versions that use neural network factors (FitzGerald et al., 2015). Note that our sequential model in this setting is essentially the same as the DB-LSTM model (Zhou and Xu, 2015) since all the frame-specific constraints are removed, except that we use simpler input features.[10] We observe a similar performance trend among our models. However, the performance gain introduced by our integrated model is relatively small compared to our FrameNet results. Note that the argument structures in CoNLL 2005 is much simpler and less diverse than the ones in FrameNet. This may lead to less complementary information captured by the sequential model and the relational model. Overall, our integrated model achieves comparable performance to the previously published results.

7.6 Analysis

We perform further analysis of our results on FrameNet to better understand our models.

We first look at how well our models perform on sentences of different lengths. In general, longer sentences tend to have more predicates and are more likely to contain complex long-range predicate-argument dependencies. We divide the FrameNet test set into 7 bins based on sentence lengths, each with length increased by 10, and the last bin contains sentences of length > 60. Figure 3 shows the F1 scores for full structure extraction for each bin. For all our models, performance tends to degrade as sentence length increases. Interestingly, our relational model consistently outperforms our sequential model at different sentence lengths, which demonstrates its robustness of handling relations of different ranges. The combination of the two models leads to consistent performance gains, and our final joint model performs the best across different sentence lengths.

Next, we analyze the errors made by different models. In general, our sequential model produces higher recall than the relational model and the integrated model, but it has lower precision. For example, for the first sentence in Figure 4, the sequential model mistakenly predicts "by $50 million" as a means to earn while both the relational and integrated models avoid this mistake. This

[10] Our reimplementation using the same feature set as Zhou and Xu (2015) did not achieve the same performance, see § 4.1 for details.

shows that performing sequential predictions over individual words has limitations. Although our relational models are good at reducing precision errors, they can be affected by frame identification errors if they are used in a pipeline. This is demonstrated by the second sentence in Figure 4, where only the *JointAll* model correctly predicts that the word "train" triggers a "Vehicle" frame. All the pipeline approaches mistakenly predict the "Education_teaching" frame in the first stage. In the second stage, the sequential model further extracts wrong semantic roles "Student" and "Institution". While the relational model and the integrated model extract no semantic roles, the frame prediction mistake remains.

8 Conclusion

We presented a new method for frame-semantic parsing that achieves the new state of the art results on standard FrameNet data. Our model integrates a sequential neural network into the learning of a relational neural network for more accurate span-based semantic role labeling. During inference, it jointly predicts frames and semantic roles using a graphical model with neural network factors. Empirical results demonstrate that our approach significantly outperforms existing neural and non-neural approaches on FrameNet data. Our model can also be adapted to perform PropBank-style SRL and it demonstrates comparable performance with the state of the art on CoNLL 2005 data.

Acknowledgments

This research was supported in part by DARPA under contract number FA8750-13-2-0005, and by NSF grants IIS-1065251 and IIS-1247489. We also gratefully acknowledge the support of the Microsoft Azure for Research program and the AWS Cloud Credits for Research program. In addition, we would like to thank anonymous reviewers for their helpful comments.

References

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*. pages 86–90.

David Belanger, Bishan Yang, and Andrew McCallum. 2017. End-to-end learning for structured prediction energy networks. In *ICML*.

Anders Björkelund, Bernd Bohnet, Love Hafdell, and Pierre Nugues. 2010. A high-performance syntactic and semantic dependency parser. In *Proceedings of the 23rd International Conference on Computational Linguistics: Demonstrations*. Association for Computational Linguistics, pages 33–36.

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

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12(Aug):2493–2537.

Dipanjan Das. 2014. Statistical models for frame-semantic parsing. In *Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore*. volume 1929, pages 26–29.

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

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.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *ACL*.

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

Charles J Fillmore, Christopher R Johnson, and Miriam RL Petruck. 2003. Background to framenet. *International journal of lexicography* 16(3):235–250.

Nicholas FitzGerald, Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Semantic role labeling with neural network factors. In *EMNLP*. pages 960–970.

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

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and whats next. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*.

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

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

Richard Johansson and Pierre Nugues. 2007. Lth: semantic structure extraction using nonprojective dependency trees. In *Proceedings of the 4th international workshop on semantic evaluations*. Association for Computational Linguistics, pages 227–230.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *ICLR*.

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

Ding Liu and Daniel Gildea. 2010. Semantic role features for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, pages 716–724.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*. pages 55–60.

André FT Martins, Mario AT Figeuiredo, Pedro MQ Aguiar, Noah A Smith, and Eric P Xing. 2011. An augmented lagrangian approach to constrained map inference. In *ICML*.

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

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. Context-aware frame-semantic role labeling. *Transactions of the Association for Computational Linguistics* 3:449–460.

Michael Roth and Mirella Lapata. 2016. Neural semantic role labeling with dependency path embeddings. In *ACL*.

Michael Roth and Kristian Woodsend. 2014. Composition of word representations improves semantic role labelling. In *EMNLP*. Citeseer, pages 407–413.

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

Swabha Swayamdipta, Miguel Ballesteros, Chris Dyer, and Noah A Smith. 2016. Greedy, joint syntactic-semantic parsing with stack lstms. In *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.

Jason Weston, Samy Bengio, and Nicolas Usunier. 2011. Wsabie: Scaling up to large vocabulary image annotation. In *IJCAI*. volume 11, pages 2764–2770.

John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2015. Towards universal paraphrastic sentence embeddings. In *ICLR*.

Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. In *EMNLP*. pages 88–94.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *ACL (1)*. pages 1127–1137.