# Linguistically-Informed Self-Attention for Semantic Role Labeling

**Emma Strubell[1], Patrick Verga[1], Daniel Andor[2], David Weiss[2] and Andrew McCallum[1]**
[1]College of Information and Computer Sciences
University of Massachusetts Amherst
`{strubell, pat, mccallum}@cs.umass.edu`
[2]Google AI Language
New York, NY
`{andor, djweiss}@google.com`

## Abstract

Current state-of-the-art semantic role labeling (SRL) uses a deep neural network with no explicit linguistic features. However, prior work has shown that gold syntax trees can dramatically improve SRL decoding, suggesting the possibility of increased accuracy from explicit modeling of syntax. In this work, we present linguistically-informed self-attention (LISA): a neural network model that combines multi-head self-attention with multi-task learning across dependency parsing, part-of-speech tagging, predicate detection and SRL. Unlike previous models which require significant pre-processing to prepare linguistic features, LISA can incorporate syntax using merely raw tokens as input, encoding the sequence only once to simultaneously perform parsing, predicate detection and role labeling for all predicates. Syntax is incorporated by training one attention head to attend to syntactic parents for each token. Moreover, if a high-quality syntactic parse is already available, it can be beneficially injected at test time without re-training our SRL model. In experiments on CoNLL-2005 SRL, LISA achieves new state-of-the-art performance for a model using predicted predicates and standard word embeddings, attaining 2.5 F1 absolute higher than the previous state-of-the-art on newswire and more than 3.5 F1 on out-of-domain data, nearly 10% reduction in error. On ConLL-2012 English SRL we also show an improvement of more than 2.5 F1. LISA also out-performs the state-of-the-art with contextually-encoded (ELMo) word representations, by nearly 1.0 F1 on news and more than 2.0 F1 on out-of-domain text.

## 1 Introduction

Semantic role labeling (SRL) extracts a high-level representation of meaning from a sentence, labeling e.g. *who* did *what* to *whom*. Explicit representations of such semantic information have been shown to improve results in challenging downstream tasks such as dialog systems (Tur et al., 2005; Chen et al., 2013), machine reading (Berant et al., 2014; Wang et al., 2015) and translation (Liu and Gildea, 2010; Bazrafshan and Gildea, 2013).

Though syntax was long considered an obvious prerequisite for SRL systems (Levin, 1993; Punyakanok et al., 2008), recently deep neural network architectures have surpassed syntactically-informed models (Zhou and Xu, 2015; Marcheggiani et al., 2017; He et al., 2017; Tan et al., 2018; He et al., 2018), achieving state-of-the art SRL performance with no explicit modeling of syntax. An additional benefit of these end-to-end models is that they require just raw tokens and (usually) detected predicates as input, whereas richer linguistic features typically require extraction by an auxiliary pipeline of models.

Still, recent work (Roth and Lapata, 2016; He et al., 2017; Marcheggiani and Titov, 2017) indicates that neural network models could see even higher accuracy gains by leveraging syntactic information rather than ignoring it. He et al. (2017) indicate that many of the errors made by a syntax-free neural network on SRL are tied to certain syntactic confusions such as prepositional phrase attachment, and show that while constrained inference using a relatively low-accuracy predicted parse can provide small improvements in SRL accuracy, providing a gold-quality parse leads to substantial gains. Marcheggiani and Titov (2017) incorporate syntax from a high-quality parser (Kiperwasser and Goldberg, 2016) using graph convolutional neural networks (Kipf and Welling, 2017), but like He et al. (2017) they attain only small increases over a model with no syntactic parse, and even perform worse than a syntax-free model on out-of-domain data. These works suggest that though syntax has the potential to improve neural network SRL models, we have not

yet designed an architecture which maximizes the benefits of auxiliary syntactic information.

In response, we propose *linguistically-informed self-attention* (LISA): a model that combines multi-task learning (Caruana, 1993) with stacked layers of multi-head self-attention (Vaswani et al., 2017); the model is trained to: (1) jointly predict parts of speech and predicates; (2) perform parsing; and (3) attend to syntactic parse parents, while (4) assigning semantic role labels. Whereas prior work typically requires separate models to provide linguistic analysis, including most syntax-free neural models which still rely on external predicate detection, our model is truly end-to-end: earlier layers are trained to predict prerequisite parts-of-speech and predicates, the latter of which are supplied to later layers for scoring. Though prior work re-encodes each sentence to predict each desired task and again with respect to each predicate to perform SRL, we more efficiently encode each sentence only once, predict its predicates, part-of-speech tags and labeled syntactic parse, then predict the semantic roles for all predicates in the sentence in parallel. The model is trained such that, as syntactic parsing models improve, providing high-quality parses at test time will improve its performance, allowing the model to leverage updated parsing models without requiring re-training.

In experiments on the CoNLL-2005 and CoNLL-2012 datasets we show that our linguistically-informed models out-perform the syntax-free state-of-the-art. On CoNLL-2005 with predicted predicates and standard word embeddings, our single model out-performs the previous state-of-the-art model on the WSJ test set by 2.5 F1 points absolute. On the challenging out-of-domain Brown test set, our model improves substantially over the previous state-of-the-art by more than 3.5 F1, a nearly 10% reduction in error. On CoNLL-2012, our model gains more than 2.5 F1 absolute over the previous state-of-the-art. Our models also show improvements when using contextually-encoded word representations (Peters et al., 2018), obtaining nearly 1.0 F1 higher than the state-of-the-art on CoNLL-2005 news and more than 2.0 F1 improvement on out-of-domain text.[1]
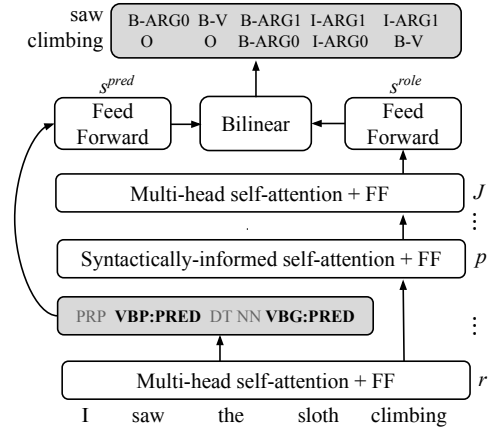
Figure 1: Word embeddings are input to $J$ layers of multi-head self-attention. In layer $p$ one attention head is trained to attend to parse parents (Figure 2). Layer $r$ is input for a joint predicate/POS classifier. Representations from layer $r$ corresponding to predicted predicates are passed to a bilinear operation scoring distinct predicate and role representations to produce per-token SRL predictions with respect to each predicted predicate.

## 2 Model

Our goal is to design an efficient neural network model which makes use of linguistic information as effectively as possible in order to perform end-to-end SRL. LISA achieves this by combining: (1) A new technique of supervising neural attention to predict syntactic dependencies with (2) multi-task learning across four related tasks.

Figure 1 depicts the overall architecture of our model. The basis for our model is the Transformer encoder introduced by Vaswani et al. (2017): we transform word embeddings into contextually-encoded token representations using stacked multi-head self-attention and feed-forward layers (§2.1).

To incorporate syntax, one self-attention head is trained to attend to each token's syntactic parent, allowing the model to use this attention head as an oracle for syntactic dependencies. We introduce this *syntactically-informed self-attention* (Figure 2) in more detail in §2.2.

Our model is designed for the more realistic setting in which gold predicates are not provided at test-time. Our model predicts predicates and integrates part-of-speech (POS) information into earlier layers by re-purposing representations closer to the input to predict predicate and POS tags us-
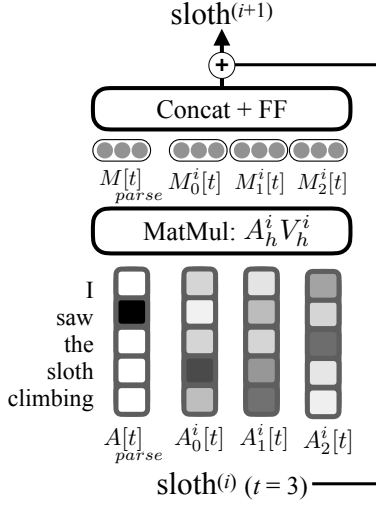
sloth$^{(i+1)}$

Concat + FF

$M[t] \atop parse$  $M_0^i[t]$ $M_1^i[t]$ $M_2^i[t]$

MatMul: $A_h^i V_h^i$

I
saw
the
sloth
climbing

$A[t] \atop parse$  $A_0^i[t]$  $A_1^i[t]$  $A_2^i[t]$

sloth$^{(i)}$ ($t = 3$)

Figure 2: Syntactically-informed self-attention for the query word *sloth*. Attention weights $A_{parse}$ heavily weight the token's syntactic governor, *saw*, in a weighted average over the token values $V_{parse}$. The other attention heads act as usual, and the attended representations from all heads are concatenated and projected through a feed-forward layer to produce the syntactically-informed representation for *sloth*.

ing hard parameter sharing (§2.3). We simplify optimization and benefit from shared statistical strength derived from highly correlated POS and predicates by treating tagging and predicate detection as a single task, performing multi-class classification into the joint Cartesian product space of POS and predicate labels.

Though typical models, which re-encode the sentence for each predicate, can simplify SRL to token-wise tagging, our joint model requires a different approach to classify roles with respect to each predicate. Contextually encoded tokens are projected to distinct *predicate* and *role* embeddings (§2.4), and each predicted predicate is scored with the sequence's role representations using a bilinear model (Eqn. 6), producing per-label scores for BIO-encoded semantic role labels for each token and each semantic frame.

The model is trained end-to-end by maximum likelihood using stochastic gradient descent (§2.5).

## 2.1 Self-attention token encoder

The basis for our model is a multi-head self-attention token encoder, recently shown to achieve state-of-the-art performance on SRL (Tan et al., 2018), and which provides a natural mechanism

for incorporating syntax, as described in §2.2. Our implementation replicates Vaswani et al. (2017).

The input to the network is a sequence $\mathcal{X}$ of $T$ token representations $x_t$. In the standard setting these token representations are initialized to pre-trained word embeddings, but we also experiment with supplying pre-trained ELMo representations combined with task-specific learned parameters, which have been shown to substantially improve performance of other SRL models (Peters et al., 2018). For experiments with gold predicates, we concatenate a predicate indicator embedding $p_t$ following previous work (He et al., 2017).

We project[2] these input embeddings to a representation that is the same size as the output of the self-attention layers. We then add a positional encoding vector computed as a deterministic sinusoidal function of $t$, since the self-attention has no innate notion of token position.

We feed this token representation as input to a series of $J$ residual multi-head self-attention layers with feed-forward connections. Denoting the $j$th self-attention layer as $T^{(j)}(\cdot)$, the output of that layer $s_t^{(j)}$, and $LN(\cdot)$ layer normalization, the following recurrence applied to initial input $c_t^{(p)}$:

$$s_t^{(j)} = LN(s_t^{(j-1)} + T^{(j)}(s_t^{(j-1)})) \qquad (1)$$

gives our final token representations $s_t^{(j)}$. Each $T^{(j)}(\cdot)$ consists of: (a) multi-head self-attention and (b) a feed-forward projection.

The multi-head self attention consists of $H$ attention heads, each of which learns a distinct attention function to attend to all of the tokens in the sequence. This self-attention is performed for each token for each head, and the results of the $H$ self-attentions are concatenated to form the final self-attended representation for each token.

Specifically, consider the matrix $S^{(j-1)}$ of $T$ token representations at layer $j - 1$. For each attention head $h$, we project this matrix into distinct key, value and query representations $K_h^{(j)}$, $V_h^{(j)}$ and $Q_h^{(j)}$ of dimensions $T \times d_k$, $T \times d_q$, and $T \times d_v$, respectively. We can then multiply $Q_h^{(j)}$ by $K_h^{(j)}$ to obtain a $T \times T$ matrix of attention weights $A_h^{(j)}$ between each pair of tokens in the sentence. Following Vaswani et al. (2017) we perform scaled dot-product attention: We scale the weights by the inverse square root of their embedding dimension

---

[2]All linear projections include bias terms, which we omit in this exposition for the sake of clarity.

and normalize with the softmax function to produce a distinct distribution for each token over all the tokens in the sentence:

$$A_h^{(j)} = \text{softmax}(d_k^{-0.5} Q_h^{(j)} K_h^{(j)T}) \qquad (2)$$

These attention weights are then multiplied by $V_h^{(j)}$ for each token to obtain the self-attended token representations $M_h^{(j)}$:

$$M_h^{(j)} = A_h^{(j)} V_h^{(j)} \qquad (3)$$

Row $t$ of $M_h^{(j)}$, the self-attended representation for token $t$ at layer $j$, is thus the weighted sum with respect to $t$ (with weights given by $A_h^{(j)}$) over the token representations in $V_h^{(j)}$.

The outputs of all attention heads for each token are concatenated, and this representation is passed to the feed-forward layer, which consists of two linear projections each followed by leaky ReLU activations (Maas et al., 2013). We add the output of the feed-forward to the initial representation and apply layer normalization to give the final output of self-attention layer $j$, as in Eqn. 1.

## 2.2 Syntactically-informed self-attention

Typically, neural attention mechanisms are left on their own to learn to attend to relevant inputs. Instead, we propose training the self-attention to attend to specific tokens corresponding to the syntactic structure of the sentence as a mechanism for passing linguistic knowledge to later layers.

Specifically, we replace one attention head with the deep bi-affine model of Dozat and Manning (2017), trained to predict syntactic dependencies. Let $A_{parse}$ be the parse attention weights, at layer $i$. Its input is the matrix of token representations $S^{(i-1)}$. As with the other attention heads, we project $S^{(i-1)}$ into key, value and query representations, denoted $K_{parse}, Q_{parse}, V_{parse}$. Here the key and query projections correspond to *parent* and *dependent* representations of the tokens, and we allow their dimensions to differ from the rest of the attention heads to more closely follow the implementation of Dozat and Manning (2017). Unlike the other attention heads which use a dot product to score key-query pairs, we score the compatibility between $K_{parse}$ and $Q_{parse}$ using a bi-affine operator $U_{heads}$ to obtain attention weights:

$$A_{parse} = \text{softmax}(Q_{parse} U_{heads} K_{parse}^T) \qquad (4)$$

These attention weights are used to compose a weighted average of the value representations $V_{parse}$ as in the other attention heads.

We apply auxiliary supervision at this attention head to encourage it to attend to each token's parent in a syntactic dependency tree, and to encode information about the token's dependency label. Denoting the attention weight from token $t$ to a candidate head $q$ as $A_{parse}[t, q]$, we model the probability of token $t$ having parent $q$ as:

$$P(q = \text{head}(t) \mid \mathcal{X}) = A_{parse}[t, q] \qquad (5)$$

using the attention weights $A_{parse}[t]$ as the distribution over possible heads for token $t$. We define the root token as having a self-loop. This attention head thus emits a directed graph[3] where each token's parent is the token to which the attention $A_{parse}$ assigns the highest weight.

We also predict dependency labels using per-class bi-affine operations between parent and dependent representations $Q_{parse}$ and $K_{parse}$ to produce per-label scores, with locally normalized probabilities over dependency labels $y_t^{dep}$ given by the softmax function. We refer the reader to Dozat and Manning (2017) for more details.

This attention head now becomes an oracle for syntax, denoted $\mathcal{P}$, providing a dependency parse to downstream layers. This model not only predicts its own dependency arcs, but allows for the injection of auxiliary parse information at test time by simply setting $A_{parse}$ to the parse parents produced by e.g. a state-of-the-art parser. In this way, our model can benefit from improved, external parsing models without re-training. Unlike typical multi-task models, ours maintains the ability to leverage external syntactic information.

## 2.3 Multi-task learning

We also share the parameters of lower layers in our model to predict POS tags and predicates. Following He et al. (2017), we focus on the end-to-end setting, where predicates must be predicted on-the-fly. Since we also train our model to predict syntactic dependencies, it is beneficial to give the model knowledge of POS information. While much previous work employs a pipelined approach to both POS tagging for dependency parsing and predicate detection for SRL, we take a multi-task learning (MTL) approach (Caruana,

---

[3]Usually the head emits a tree, but we do not enforce it here.

1993), sharing the parameters of earlier layers in our SRL model with a joint POS and predicate detection objective. Since POS is a strong predictor of predicates[4] and the complexity of training a multi-task model increases with the number of tasks, we combine POS tagging and predicate detection into a joint label space: For each POS tag TAG which is observed co-occurring with a predicate, we add a label of the form TAG:PREDICATE.

Specifically, we feed the representation $s_t^{(r)}$ from a layer $r$ preceding the syntactically-informed layer $p$ to a linear classifier to produce per-class scores $r_t$ for token $t$. We compute locally-normalized probabilities using the softmax function: $P(y_t^{prp} \mid \mathcal{X}) \propto \exp(r_t)$, where $y_t^{prp}$ is a label in the joint space.

## 2.4 Predicting semantic roles

Our final goal is to predict semantic roles for each predicate in the sequence. We score each predicate against each token in the sequence using a bilinear operation, producing per-label scores for each token for each predicate, with predicates and syntax determined by oracles $\mathcal{V}$ and $\mathcal{P}$.

First, we project each token representation $s_t^{(J)}$ to a predicate-specific representation $s_t^{pred}$ and a role-specific representation $s_t^{role}$. We then provide these representations to a bilinear transformation $U$ for scoring. So, the role label scores $s_{ft}$ for the token at index $t$ with respect to the predicate at index $f$ (i.e. token $t$ and frame $f$) are given by:

$$s_{ft} = (s_f^{pred})^T U s_t^{role} \qquad (6)$$

which can be computed in parallel across all semantic frames in an entire minibatch. We calculate a locally normalized distribution over role labels for token $t$ in frame $f$ using the softmax function: $P(y_{ft}^{role} \mid \mathcal{P}, \mathcal{V}, \mathcal{X}) \propto \exp(s_{ft})$.

At test time, we perform constrained decoding using the Viterbi algorithm to emit valid sequences of BIO tags, using unary scores $s_{ft}$ and the transition probabilities given by the training data.

## 2.5 Training

We maximize the sum of the likelihoods of the individual tasks. In order to maximize our model's ability to leverage syntax, during training we clamp $\mathcal{P}$ to the gold parse ($\mathcal{P}_G$) and $\mathcal{V}$ to gold predicates $\mathcal{V}_G$ when passing parse and predicate

---

[4]All predicates in CoNLL-2005 are verbs; CoNLL-2012 includes some nominal predicates.

representations to later layers, whereas syntactic head prediction and joint predicate/POS prediction are conditioned only on the input sequence $\mathcal{X}$. The overall objective is thus:

$$\frac{1}{T} \sum_{t=1}^{T} \Big[ \sum_{f=1}^{F} \log P(y_{ft}^{role} \mid \mathcal{P}_G, \mathcal{V}_G, \mathcal{X})$$
$$+ \log P(y_t^{prp} \mid \mathcal{X})$$
$$+ \lambda_1 \log P(\text{head}(t) \mid \mathcal{X})$$
$$+ \lambda_2 \log P(y_t^{dep} \mid \mathcal{P}_G, \mathcal{X}) \Big] \qquad (7)$$

where $\lambda_1$ and $\lambda_2$ are penalties on the syntactic attention loss.

We train the model using Nadam (Dozat, 2016) SGD combined with the learning rate schedule in Vaswani et al. (2017). In addition to MTL, we regularize our model using dropout (Srivastava et al., 2014). We use gradient clipping to avoid exploding gradients (Bengio et al., 1994; Pascanu et al., 2013). Additional details on optimization and hyperparameters are included in Appendix A.

## 3 Related work

Early approaches to SRL (Pradhan et al., 2005; Surdeanu et al., 2007; Johansson and Nugues, 2008; Toutanova et al., 2008) focused on developing rich sets of linguistic features as input to a linear model, often combined with complex constrained inference e.g. with an ILP (Punyakanok et al., 2008). Täckström et al. (2015) showed that constraints could be enforced more efficiently using a clever dynamic program for exact inference. Sutton and McCallum (2005) modeled syntactic parsing and SRL jointly, and Lewis et al. (2015) jointly modeled SRL and CCG parsing.

Collobert et al. (2011) were among the first to use a neural network model for SRL, a CNN over word embeddings which failed to out-perform non-neural models. FitzGerald et al. (2015) successfully employed neural networks by embedding lexicalized features and providing them as factors in the model of Täckström et al. (2015).

More recent neural models are syntax-free. Zhou and Xu (2015), Marcheggiani et al. (2017) and He et al. (2017) all use variants of deep LSTMs with constrained decoding, while Tan et al. (2018) apply self-attention to obtain state-of-the-art SRL with gold predicates. Like this work, He et al. (2017) present end-to-end experiments, predicting predicates using an LSTM, and He et al.

(2018) jointly predict SRL spans and predicates in a model based on that of Lee et al. (2017), obtaining state-of-the-art predicted predicate SRL. Concurrent to this work, Peters et al. (2018) and He et al. (2018) report significant gains on PropBank SRL by training a wide LSTM language model and using a task-specific transformation of its hidden representations (ELMo) as a deep, and computationally expensive, alternative to typical word embeddings. We find that LISA obtains further accuracy increases when provided with ELMo word representations, especially on out-of-domain data.

Some work has incorporated syntax into neural models for SRL. Roth and Lapata (2016) incorporate syntax by embedding dependency paths, and similarly Marcheggiani and Titov (2017) encode syntax using a graph CNN over a predicted syntax tree, out-performing models without syntax on CoNLL-2009. These works are limited to incorporating partial dependency paths between tokens whereas our technique incorporates the entire parse. Additionally, Marcheggiani and Titov (2017) report that their model does not out-perform syntax-free models on out-of-domain data, a setting in which our technique excels.

MTL (Caruana, 1993) is popular in NLP, and others have proposed MTL models which incorporate subsets of the tasks we do (Collobert et al., 2011; Zhang and Weiss, 2016; Hashimoto et al., 2017; Peng et al., 2017; Swayamdipta et al., 2017), and we build off work that investigates where and when to combine different tasks to achieve the best results (Søgaard and Goldberg, 2016; Bingel and Søgaard, 2017; Alonso and Plank, 2017). Our specific method of incorporating supervision into self-attention is most similar to the concurrent work of Liu and Lapata (2018), who use edge marginals produced by the matrix-tree algorithm as attention weights for document classification and natural language inference.

The question of training on gold versus predicted labels is closely related to learning to search (Daumé III et al., 2009; Ross et al., 2011; Chang et al., 2015) and scheduled sampling (Bengio et al., 2015), with applications in NLP to sequence labeling and transition-based parsing (Choi and Palmer, 2011; Goldberg and Nivre, 2012; Ballesteros et al., 2016). Our approach may be interpreted as an extension of teacher forcing (Williams and Zipser, 1989) to MTL. We leave exploration of more advanced scheduled sampling techniques to future work.

# 4 Experimental results

We present results on the CoNLL-2005 shared task (Carreras and Màrquez, 2005) and the CoNLL-2012 English subset of OntoNotes 5.0 (Pradhan et al., 2006), achieving state-of-the-art results for a single model with predicted predicates on both corpora. We experiment with both standard pre-trained GloVe word embeddings (Pennington et al., 2014) and pre-trained ELMo representations with fine-tuned task-specific parameters (Peters et al., 2018) in order to best compare to prior work. Hyperparameters that resulted in the best performance on the validation set were selected via a small grid search, and models were trained for a maximum of 4 days on one TitanX GPU using early stopping on the validation set. We convert constituencies to dependencies using the Stanford head rules v3.5 (de Marneffe and Manning, 2008). A detailed description of hyperparameter settings and data pre-processing can be found in Appendix A.

We compare our **LISA** models to four strong baselines: For experiments using predicted predicates, we compare to He et al. (2018) and the ensemble model (**PoE**) from He et al. (2017), as well as a version of our own self-attention model which does not incorporate syntactic information (**SA**). To compare to more prior work, we present additional results on CoNLL-2005 with models given gold predicates at test time. In these experiments we also compare to Tan et al. (2018), the previous state-of-the art SRL model using gold predicates and standard embeddings.

We demonstrate that our models benefit from injecting state-of-the-art predicted parses at test time (**+D&M**) by fixing the attention to parses predicted by Dozat and Manning (2017), the winner of the 2017 CoNLL shared task (Zeman et al., 2017) which we re-train using ELMo embeddings. In all cases, using these parses at test time improves performance.

We also evaluate our model using the gold syntactic parse at test time (**+Gold**), to provide an upper bound for the benefit that syntax could have for SRL using LISA. These experiments show that despite LISA's strong performance, there remains substantial room for improvement. In §4.3 we perform further analysis comparing SRL models using gold and predicted parses.

| GloVe | Dev | | | WSJ Test | | | Brown Test | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 |
| He et al. (2017) PoE | 81.8 | 81.2 | 81.5 | 82.0 | 83.4 | 82.7 | 69.7 | 70.5 | 70.1 |
| He et al. (2018) | 81.3 | 81.9 | 81.6 | 81.2 | 83.9 | 82.5 | 69.7 | 71.9 | 70.8 |
| SA | 83.52 | 81.28 | 82.39 | 84.17 | 83.28 | 83.72 | 72.98 | 70.1 | 71.51 |
| LISA | 83.1 | 81.39 | 82.24 | 84.07 | 83.16 | 83.61 | 73.32 | 70.56 | 71.91 |
| +D&M | **84.59** | **82.59** | **83.58** | **85.53** | **84.45** | **84.99** | **75.8** | **73.54** | **74.66** |
| +Gold | *87.91* | *85.73* | *86.81* | — | — | — | — | — | — |
| | | | | | | | | | |
| ELMo | | | | | | | | | |
| He et al. (2018) | 84.9 | **85.7** | 85.3 | 84.8 | **87.2** | 86.0 | 73.9 | **78.4** | 76.1 |
| SA | 85.78 | 84.74 | 85.26 | 86.21 | 85.98 | 86.09 | 77.1 | 75.61 | 76.35 |
| LISA | **86.07** | 84.64 | **85.35** | 86.69 | 86.42 | 86.55 | 78.95 | 77.17 | 78.05 |
| +D&M | 85.83 | 84.51 | 85.17 | **87.13** | 86.67 | **86.90** | **79.02** | 77.49 | **78.25** |
| +Gold | *88.51* | *86.77* | *87.63* | — | — | — | — | — | — |

Table 1: Precision, recall and F1 on the CoNLL-2005 development and test sets.

| WSJ Test | P | R | F1 |
|---|---|---|---|
| He et al. (2018) | 84.2 | 83.7 | 83.9 |
| Tan et al. (2018) | 84.5 | 85.2 | 84.8 |
| SA | 84.7 | 84.24 | 84.47 |
| LISA | 84.72 | 84.57 | 84.64 |
| +D&M | **86.02** | **86.05** | **86.04** |
| | | | |
| Brown Test | P | R | F1 |
| He et al. (2018) | 74.2 | 73.1 | 73.7 |
| Tan et al. (2018) | 73.5 | 74.6 | 74.1 |
| SA | 73.89 | 72.39 | 73.13 |
| LISA | 74.77 | 74.32 | 74.55 |
| +D&M | **76.65** | **76.44** | **76.54** |

Table 2: Precision, recall and F1 on CoNLL-2005 with gold predicates.

## 4.1 Semantic role labeling

Table 1 lists precision, recall and F1 on the CoNLL-2005 development and test sets using predicted predicates. For models using GloVe embeddings, our syntax-free SA model already achieves a new state-of-the-art by jointly predicting predicates, POS and SRL. LISA with its own parses performs comparably to SA, but when supplied with D&M parses LISA out-performs the previous state-of-the-art by 2.5 F1 points. On the out-of-domain Brown test set, LISA also performs comparably to its syntax-free counterpart with its own parses, but with D&M parses LISA performs exceptionally well, more than 3.5 F1 points higher than He et al. (2018). Incorporating ELMo em-

beddings improves all scores. The gap in SRL F1 between models using LISA and D&M parses is smaller due to LISA's improved parsing accuracy (see §4.2), but LISA with D&M parses still achieves the highest F1: nearly 1.0 absolute F1 higher than the previous state-of-the-art on WSJ, and more than 2.0 F1 higher on Brown. In both settings LISA leverages domain-agnostic syntactic information rather than over-fitting to the newswire training data which leads to high performance even on out-of-domain text.

To compare to more prior work we also evaluate our models in the artificial setting where gold predicates are provided at test time. For fair comparison we use GloVe embeddings, provide predicate indicator embeddings on the input and re-encode the sequence relative to each gold predicate. Here LISA still excels: with D&M parses, LISA out-performs the previous state-of-the-art by more than 2 F1 on both WSJ and Brown.

Table 3 reports precision, recall and F1 on the CoNLL-2012 test set. We observe performance similar to that observed on ConLL-2005: Using GloVe embeddings our SA baseline already out-performs He et al. (2018) by nearly 1.5 F1. With its own parses, LISA slightly under-performs our syntax-free model, but when provided with stronger D&M parses LISA out-performs the state-of-the-art by more than 2.5 F1. Like CoNLL-2005, ELMo representations improve all models and close the F1 gap between models supplied with LISA and D&M parses. On this dataset ELMo also substantially narrows the

| Dev | P | R | F1 |
|---|---|---|---|
| GloVe | | | |
| He et al. (2018) | 79.2 | 79.7 | 79.4 |
| SA | 82.32 | 79.76 | 81.02 |
| LISA | 81.77 | 79.65 | 80.70 |
| +D&M | **82.97** | **81.14** | **82.05** |
| +Gold | *87.57* | *85.32* | *86.43* |
| | | | |
| ELMo | | | |
| He et al. (2018) | 82.1 | **84.0** | 83.0 |
| SA | 84.35 | 82.14 | 83.23 |
| LISA | **84.19** | 82.56 | **83.37** |
| +D&M | 84.09 | 82.65 | 83.36 |
| +Gold | *88.22* | *86.53* | *87.36* |

| Test | P | R | F1 |
|---|---|---|---|
| GloVe | | | |
| He et al. (2018) | 79.4 | 80.1 | 79.8 |
| SA | 82.55 | 80.02 | 81.26 |
| LISA | 81.86 | 79.56 | 80.70 |
| +D&M | **83.3** | **81.38** | **82.33** |
| | | | |
| ELMo | | | |
| He et al. (2018) | 81.9 | **84.0** | 82.9 |
| SA | **84.39** | 82.21 | 83.28 |
| LISA | 83.97 | 82.29 | 83.12 |
| +D&M | 84.14 | 82.64 | **83.38** |

Table 3: Precision, recall and F1 on the CoNLL-2012 development and test sets. Italics indicate a synthetic upper bound obtained by providing a gold parse at test time.

difference between models with- and without syntactic information. This suggests that for this challenging dataset, ELMo already encodes much of the information available in the D&M parses. Yet, higher accuracy parses could still yield improvements since providing gold parses increases F1 by 4 points even with ELMo embeddings.

### 4.2 Parsing, POS and predicate detection

We first report the labeled and unlabeled attachment scores (LAS, UAS) of our parsing models on the CoNLL-2005 and 2012 test sets (Table 4) with GloVe ($G$) and ELMo ($E$) embeddings. D&M achieves the best scores. Still, LISA's GloVe UAS is comparable to popular off-the-shelf dependency parsers such as spaCy,[5] and with ELMo

---
[5] spaCy reports 94.48 UAS on WSJ using Stanford dependencies v3.3: https://spacy.io/usage/

| Data | Model | POS | UAS | LAS |
|---|---|---|---|---|
| | $D\&M_E$ | — | 96.48 | 94.40 |
| WSJ | $LISA_G$ | 96.92 | 94.92 | 91.87 |
| | $LISA_E$ | 97.80 | 96.28 | 93.65 |
| | $D\&M_E$ | — | 92.56 | 88.52 |
| Brown | $LISA_G$ | 94.26 | 90.31 | 85.82 |
| | $LISA_E$ | 95.77 | 93.36 | 88.75 |
| | $D\&M_E$ | — | 94.99 | 92.59 |
| CoNLL-12 | $LISA_G$ | 96.81 | 93.35 | 90.42 |
| | $LISA_E$ | 98.11 | 94.84 | 92.23 |

Table 4: Parsing (labeled and unlabeled attachment) and POS accuracies attained by the models used in SRL experiments on test datasets. Subscript $G$ denotes GloVe and $E$ ELMo embeddings.

| | Model | P | R | F1 |
|---|---|---|---|---|
| WSJ | He et al. (2017) | 94.5 | 98.5 | 96.4 |
| | LISA | 98.9 | 97.9 | 98.4 |
| Brown | He et al. (2017) | 89.3 | 95.7 | 92.4 |
| | LISA | 95.5 | 91.9 | 93.7 |
| CoNLL-12 | LISA | 99.8 | 94.7 | 97.2 |

Table 5: Predicate detection precision, recall and F1 on CoNLL-2005 and CoNLL-2012 test sets.

embeddings comparable to the standalone D&M parser. The difference in parse accuracy between $LISA_G$ and D&M likely explains the large increase in SRL performance we see from decoding with D&M parses in that setting.

In Table 5 we present predicate detection precision, recall and F1 on the CoNLL-2005 and 2012 test sets. SA and LISA with and without ELMo attain comparable scores so we report only LISA+GloVe. We compare to He et al. (2017) on CoNLL-2005, the only cited work reporting comparable predicate detection F1. LISA attains high predicate detection scores, above 97 F1, on both in-domain datasets, and out-performs He et al. (2017) by 1.5-2 F1 points even on the out-of-domain Brown test set, suggesting that multi-task learning works well for SRL predicate detection.

### 4.3 Analysis

First we assess SRL F1 on sentences divided by parse accuracy. Table 6 lists average SRL F1 (across sentences) for the four conditions of LISA and D&M parses being correct or not (**L**±, **D**±). Both parsers are correct on 26% of sentences.

---
facts-figures

| | L+/D+ | L–/D+ | L+/D– | L–/D– |
|---|---|---|---|---|
| Proportion | 26% | 12% | 4% | 56% |
| SA | 79.29 | 75.14 | 75.97 | 75.08 |
| LISA | 79.51 | 74.33 | 79.69 | 75.00 |
| +D&M | 79.03 | 76.96 | 77.73 | 76.52 |
| *+Gold* | *79.61* | *78.38* | *81.41* | *80.47* |

Table 6: Average SRL F1 on CoNLL-2005 for sentences where LISA (L) and D&M (D) parses were completely correct (+) or incorrect (–).
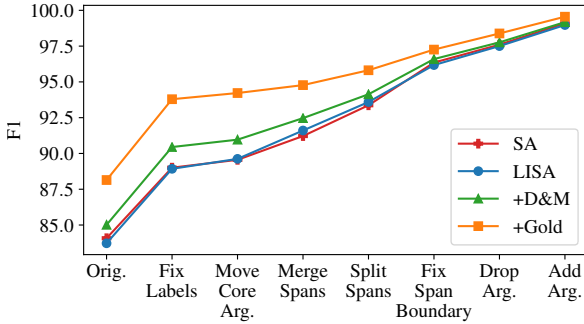


Figure 3: Performance of CoNLL-2005 models after performing corrections from He et al. (2017).

Here there is little difference between any of the models, with LISA models tending to perform slightly better than SA. Both parsers make mistakes on the majority of sentences (57%), difficult sentences where SA also performs the worst. These examples are likely where gold and D&M parses improve the most over other models in overall F1: Though both parsers fail to correctly parse the entire sentence, the D&M parser is less wrong (87.5 vs. 85.7 average LAS), leading to higher SRL F1 by about 1.5 average F1.

Following He et al. (2017), we next apply a series of corrections to model predictions in order to understand which error types the gold parse resolves: e.g. *Fix Labels* fixes labels on spans matching gold boundaries, and *Merge Spans* merges adjacent predicted spans into a gold span.[6]

In Figure 3 we see that much of the performance gap between the gold and predicted parses is due to span boundary errors (*Merge Spans*, *Split Spans* and *Fix Span Boundary*), which supports the hypothesis proposed by He et al. (2017) that incorporating syntax could be particularly helpful for resolving these errors. He et al. (2017) also point out

---

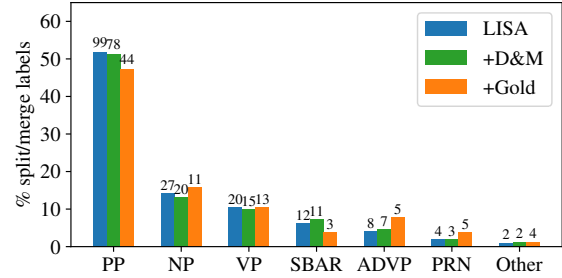[6]Refer to He et al. (2017) for a detailed explanation of the different error types.



Figure 4: Percent and count of split/merge corrections performed in Figure 3, by phrase type.

that these errors are due mainly to prepositional phrase (PP) attachment mistakes. We also find this to be the case: Figure 4 shows a breakdown of split/merge corrections by phrase type. Though the number of corrections decreases substantially across phrase types, the proportion of corrections attributed to PPs remains the same (approx. 50%) even after providing the correct PP attachment to the model, indicating that PP span boundary mistakes are a fundamental difficulty for SRL.

## 5   Conclusion

We present linguistically-informed self-attention: a multi-task neural network model that effectively incorporates rich linguistic information for semantic role labeling. LISA out-performs the state-of-the-art on two benchmark SRL datasets, including out-of-domain. Future work will explore improving LISA's parsing accuracy, developing better training techniques and adapting to more tasks.

## Acknowledgments

# References

Martın Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2015. Tensorflow: Large-scale machine learning on heterogeneous systems, 2015. *Software available from tensorflow.org*.

Héctor Martínez Alonso and Barbara Plank. 2017. When is multitask learning effective? semantic sequence prediction under varying data conditions. In *EACL*.

Miguel Ballesteros, Yoav Goldberg, Chris Dyer, and Noah A. Smith. 2016. Training with exploration improves a greedy stack lstm parser. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2005–2010.

Marzieh Bazrafshan and Daniel Gildea. 2013. Semantic roles for string to tree machine translation. In *ACL*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *NIPS*.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166.

Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Brad Huang, Christopher D. Manning, Abby Vander Linden, Brittany Harding, and Peter Clark. 2014. Modeling biological processes for reading comprehension. In *EMNLP*.

Joachim Bingel and Anders Søgaard. 2017. Identifying beneficial task relations for multi-task learning in deep neural networks. In *EACL*.

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

Rich Caruana. 1993. Multitask learning: a knowledge-based source of inductive bias. In *ICML*.

Kai-Wei Chang, Akshay Krishnamurthy, Alekh Agarwal, Hal Daumé III, and John Langford. 2015. Learning to search better than your teacher. In *ICML*.

Yun-Nung Chen, William Yang Wang, and Alexander I Rudnicky. 2013. Unsupervised induction and filling of semantic slots for spoken dialogue systems using frame-semantic parsing. In *Proc. of ASRU-IEEE*.

Jinho D. Choi and Martha Palmer. 2011. Getting the most out of transition-based dependency parsing. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: short papers*, pages 687–692.

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.

Hal Daumé III, John Langford, and Daniel Marcu. 2009. Search-based structured prediction. *Machine Learning*, 75(3):297–325.

Timothy Dozat. 2016. Incorporating nesterov momentum into adam. In *ICLR Workshop track*.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *ICLR*.

Nicholas FitzGerald, Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Semantic role labeling with neural network factors. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 960–970.

W. N. Francis and H. Kučera. 1964. Manual of information to accompany a standard corpus of present-day edited american english, for use with digital computers. Technical report, Department of Linguistics, Brown University, Providence, Rhode Island.

Yoav Goldberg and Joakim Nivre. 2012. A dynamic oracle for arc-eager dependency parsing. In *Proceedings of COLING 2012: Technical Papers*, pages 959–976.

Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple nlp tasks. In *Conference on Empirical Methods in Natural Language Processing*.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *ACL*.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and whats next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*.

Richard Johansson and Pierre Nugues. 2008. Dependency-based semantic role labeling of propbank. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 69–78.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference for Learning Representations (ICLR)*, San Diego, California, USA.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Thomas N. Kipf and Max Welling. 2017. Semisupervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *EMNLP*.

Beth Levin. 1993. *English verb classes and alternations: A preliminary investigation*. University of Chicago press.

Mike Lewis, Luheng He, and Luke Zettlemoyer. 2015. Joint A* CCG Parsing and Semantic Role Labeling. In *EMNLP*.

Ding Liu and Daniel Gildea. 2010. Semantic role features for machine translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*.

Yang Liu and Mirella Lapata. 2018. Learning structured text representations. *Transactions of the Association for Computational Linguistics*, 6:63–75.

Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. 2013. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, volume 30.

Diego Marcheggiani, Anton Frolov, and Ivan Titov. 2017. A simple and accurate syntax-agnostic neural model for dependency-based semantic role labeling. In *CoNLL*.

Diego Marcheggiani and Ivan Titov. 2017. Encoding sentences with graph convolutional networks for semantic role labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: The Penn TreeBank. *Computational Linguistics – Special issue on using large corpora: II*, 19(2):313–330.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The stanford typed dependencies representation. In *COLING 2008 Workshop on Cross-framework and Cross-domain Parser Evaluation*.

Yurii Nesterov. 1983. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. volume 27, pages 372–376.

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

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30 th International Conference on Machine Learning*.

Hao Peng, Sam Thomson, and Noah A. Smith. 2017. Deep multitask learning for semantic dependency parsing. In *ACL*.

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

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *NAACL*.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2006. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James Martin, and Dan Jurafsky. 2005. Semantic role labeling using different syntactic views. In *Proceedings of the Association for Computational Linguistics 43rd annual meeting (ACL)*.

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.

Stéphane Ross, Geoffrey J. Gordon, and J. Andrew Bagnell. 2011. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Michael Roth and Mirella Lapata. 2016. Neural semantic role labeling with dependency path embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1192–1202.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 231–235.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.

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.

Charles Sutton and Andrew McCallum. 2005. Joint parsing and semantic role labeling. In *CoNLL*.

Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A. Smith. 2017. Frame-semantic parsing with softmax-margin segmental rnns and a syntactic scaffold. In *arXiv:1706.09528*.

Oscar Täckström, Kuzman Ganchev, and Dipanjan Das. 2015. Efficient inference and structured learning for semantic role labeling. *TACL*, 3:29–41.

Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 2018. Deep semantic role labeling with self-attention. In *AAAI*.

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

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.

Gokhan Tur, Dilek Hakkani-Tür, and Ananlada Chotimongkol. 2005. Semi-supervised learning for spoken language understanding using semantic role labeling. In *ASRU*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *31st Conference on Neural Information Processing Systems (NIPS)*.

Hai Wang, Mohit Bansal, Kevin Gimpel, and David McAllester. 2015. Machine comprehension with syntax, frames, and semantics. In *ACL*.

R. J. Williams and D. Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, et al. 2017. Conll 2017 shared task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

Yuan Zhang and David Weiss. 2016. Stack-propagation: Improved representation learning for syntax. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1557–1566. Association for Computational Linguistics.

Jie Zhou and Wei Xu. 2015. End-to-end learning of semantic role labeling using recurrent neural networks. In *Proc. of the Annual Meeting of the Association for Computational Linguistics (ACL)*.