Jointly Predicting Predicates and Arguments in Neural Semantic Role Labeling

Luheng He **Kenton Lee** **Omer Levy** **Luke Zettlemoyer**
Paul G. Allen School of Computer Science & Engineering
University of Washington, Seattle WA
{luheng, kentonl, omerlevy, lsz}@cs.washington.edu

Abstract

Recent BIO-tagging-based neural semantic role labeling models are very high performing, but assume gold predicates as part of the input and cannot incorporate span-level features. We propose an end-to-end approach for jointly predicting all predicates, arguments spans, and the relations between them. The model makes independent decisions about what relationship, if any, holds between every possible word-span pair, and learns contextualized span representations that provide rich, shared input features for each decision. Experiments demonstrate that this approach sets a new state of the art on PropBank SRL without gold predicates.[1]

1 Introduction

Semantic role labeling (SRL) captures predicate-argument relations, such as "who did what to whom." Recent high-performing SRL models (He et al., 2017; Marcheggiani et al., 2017; Tan et al., 2018) are BIO-taggers, labeling argument spans for a single predicate at a time (as shown in Figure 1). They are typically only evaluated with gold predicates, and must be pipelined with error-prone predicate identification models for deployment.

We propose an end-to-end approach for predicting all the predicates and their argument spans in one forward pass. Our model builds on a recent coreference resolution model (Lee et al., 2017), by making central use of learned, contextualized span representations. We use these representations to predict SRL graphs directly over text spans. Each edge is identified by independently predicting which role, if any, holds between every possible pair of text spans, while using aggressive beam pruning for efficiency. The final graph is simply the union of predicted SRL roles (edges) and their associated text spans (nodes).

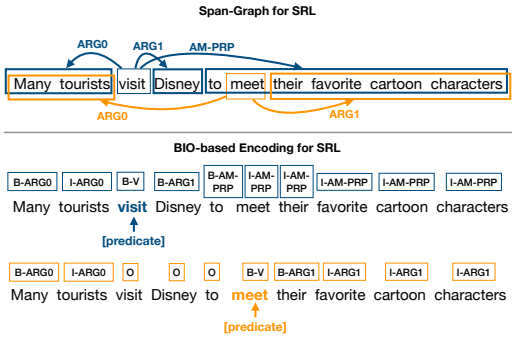

Figure 1: A comparison of our span-graph structure (top) versus BIO-based SRL (bottom).

Our span-graph formulation overcomes a key limitation of semi-markov and BIO-based models (Kong et al., 2016; Zhou and Xu, 2015; Yang and Mitchell, 2017; He et al., 2017; Tan et al., 2018): it can model overlapping spans across different predicates in the same output structure (see Figure 1). The span representations also generalize the token-level representations in BIO-based models, letting the model dynamically decide which spans and roles to include, without using previously standard syntactic features (Punyakanok et al., 2008; FitzGerald et al., 2015).

To the best of our knowledge, this is the first span-based SRL model that does not assume that predicates are given. In this more realistic setting, where the predicate must be predicted, our model achieves state-of-the-art performance on PropBank. It also reinforces the strong performance of similar span embedding methods for coreference (Lee et al., 2017), suggesting that this style of models could be used for other span-span relation tasks, such as syntactic parsing (Stern et al., 2017), relation extraction (Miwa and Bansal, 2016), and QA-SRL (FitzGerald et al., 2018).

[1] Code and models: https://github.com/luheng/lsgn

Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 364–369
Melbourne, Australia, July 15 - 20, 2018.

2 Model

We consider the space of possible predicates to be all the tokens in the input sentence, and the space of arguments to be all continuous spans. Our model decides what relation exists between each predicate-argument pair (including *no relation*).

Formally, given a sequence $X = w_1, \ldots, w_n$, we wish to predict a set of labeled predicate-argument relations $Y \subseteq \mathcal{P} \times \mathcal{A} \times \mathcal{L}$, where $\mathcal{P} = \{w_1, \ldots, w_n\}$ is the set of all tokens (predicates), $\mathcal{A} = \{(w_i, \ldots, w_j) \mid 1 \leq i \leq j \leq n\}$ contains all the spans (arguments), and $\mathcal{L}$ is the space of semantic role labels, including a null label ϵ indicating no relation. The final SRL output would be all the non-empty relations $\{(p, a, l) \in Y \mid l \neq \epsilon\}$.

We then define a set of random variables, where each random variable $y_{p,a}$ corresponds to a predicate $p \in \mathcal{P}$ and an argument $a \in \mathcal{A}$, taking value from the discrete label space $\mathcal{L}$. The random variables $y_{p,a}$ are conditionally independent of each other given the input X:

$$P(Y \mid X) = \prod_{p \in \mathcal{P}, a \in \mathcal{A}} P(y_{p,a} \mid X) \quad (1)$$

$$P(y_{p,a} = l \mid X) = \frac{\exp(\phi(p, a, l))}{\sum_{l' \in \mathcal{L}} \exp(\phi(p, a, l'))} \quad (2)$$

Where $\phi(p, a, l)$ is a scoring function for a possible (predicate, argument, label) combination. ϕ is decomposed into two unary scores on the predicate and the argument (defined in Section 3), as well as a label-specific score for the relation:

$$\phi(p, a, l) = \Phi_{\text{a}}(a) + \Phi_{\text{p}}(p) + \Phi^{(l)}_{\text{rel}}(a, p) \quad (3)$$

The score for the *null* label is set to a constant: $\phi(p, a, \epsilon) = 0$, similar to logistic regression.

Learning For each input X, we minimize the negative log likelihood of the gold structure Y^*:

$$\mathcal{J}(X) = -\log P(Y^* \mid X) \quad (4)$$

Beam pruning As our model deals with $O(n^2)$ possible argument spans and $O(n)$ possible predicates, it needs to consider $O(n^3|\mathcal{L}|)$ possible relations, which is computationally impractical. To overcome this issue, we define two beams B_{a} and B_{p} for storing the candidate arguments and predicates, respectively. The candidates in each beam are ranked by their unary score (Φ_{a} or Φ_{p}). The sizes of the beams are limited by $\lambda_{\text{a}} n$ and $\lambda_{\text{p}} n$. Elements that fall out of the beam do not participate in computing the edge factors $\Phi^{(l)}_{\text{rel}}$, reducing the overall number of relational factors evaluated by the model to $O(n^2|\mathcal{L}|)$. We also limit the maximum width of spans to a fixed number W (e.g. $W = 30$), further reducing the number of computed unary factors to $O(n)$.

3 Neural Architecture

Our model builds contextualized representations for argument spans a and predicate words p based on BiLSTM outputs (Figure 2) and uses feed-forward networks to compute the factor scores in $\phi(p, a, l)$ described in Section 2 (Figure 3).

Word-level contexts The bottom layer consists of pre-trained word embeddings concatenated with character-based representations, i.e. for each token w_i, we have $\mathbf{x}_i = [\textsc{WordEmb}(w_i); \textsc{CharCNN}(w_i)]$. We then contextualize each $\mathbf{x}_i$ using an m-layered bidirectional LSTM with highway connections (Zhang et al., 2016), which we denote as $\bar{\mathbf{x}}_i$.

Argument and predicate representation We build contextualized representations for all candidate arguments $a \in \mathcal{A}$ and predicates $p \in \mathcal{P}$. The argument representation contains the following: end points from the BiLSTM outputs $(\bar{\mathbf{x}}_{\textsc{Start}(a)}, \bar{\mathbf{x}}_{\textsc{End}(a)})$, a soft head word $\mathbf{x}_{\text{h}}(a)$, and embedded span width features $\mathbf{f}(a)$, similar to Lee et al. (2017). The predicate representation is simply the BiLSTM output at the position $\textsc{Index}(p)$.

$$\mathbf{g}(a) = [\bar{\mathbf{x}}_{\textsc{Start}(a)}; \bar{\mathbf{x}}_{\textsc{End}(a)}; \mathbf{x}_{\text{h}}(a); \mathbf{f}(a)] \quad (5)$$

$$\mathbf{g}(p) = \bar{\mathbf{x}}_{\textsc{Index}(p)} \quad (6)$$

The soft head representation $\mathbf{x}_{\text{h}}(a)$ is an attention mechanism over word inputs $\mathbf{x}$ in the argument span, where the weights $\mathbf{e}(a)$ are computed via a linear layer over the BiLSTM outputs $\bar{\mathbf{x}}$.

$$\mathbf{x}_{\text{h}}(a) = \mathbf{x}_{\textsc{Start}(a):\textsc{End}(a)} \mathbf{e}(s)^{\intercal} \quad (7)$$

$$\mathbf{e}(a) = \textsc{SoftMax}(\mathbf{w}_{\text{e}}^{\intercal} \bar{\mathbf{x}}_{\textsc{Start}(a):\textsc{End}(a)}) \quad (8)$$

$\mathbf{x}_{\textsc{Start}(a):\textsc{End}(a)}$ is a shorthand for stacking a list of vectors $\mathbf{x}_t$, where $\textsc{Start}(a) \leq t \leq \textsc{End}(a)$.

Scoring The scoring functions Φ are implemented with feed-forward networks based on the predicate and argument representations $\mathbf{g}$:

$$\Phi_{\text{a}}(a) = \mathbf{w}_{\text{a}}^{\intercal} \text{MLP}_{\text{a}}(\mathbf{g}(a)) \quad (9)$$

$$\Phi_{\text{p}}(p) = \mathbf{w}_{\text{p}}^{\intercal} \text{MLP}_{\text{p}}(\mathbf{g}(p)) \quad (10)$$

$$\Phi^{(l)}_{\text{rel}}(a, p) = \mathbf{w}_{\text{r}}^{(l)\intercal} \text{MLP}_{\text{r}}([\mathbf{g}(a); \mathbf{g}(p)]) \quad (11)$$

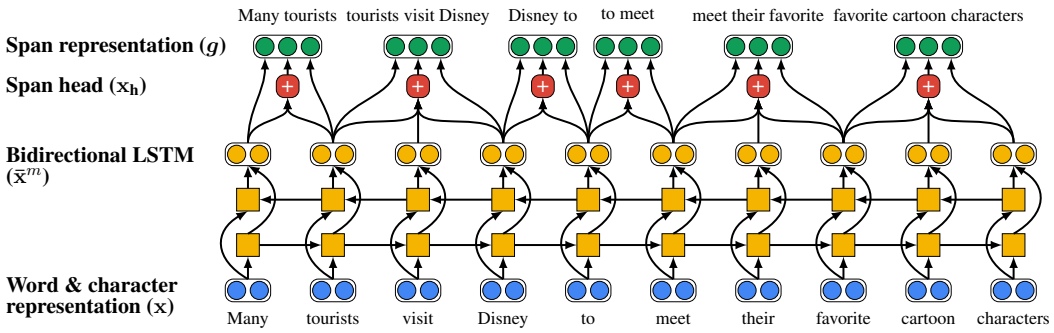

Figure 2: Building the argument span representations $\mathbf{g}(a)$ from BiLSTM outputs. For clarity, we only show one BiLSTM layer and a small subset of the arguments.

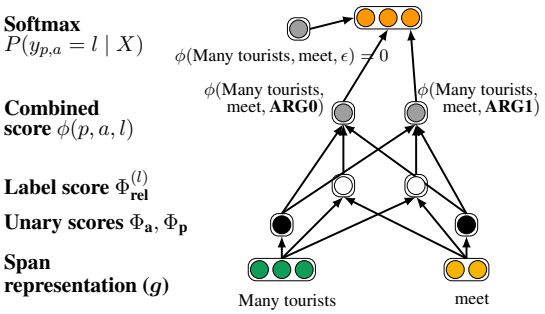

Figure 3: The span-pair classifier takes in predicate and argument representations as inputs, and computes a softmax over the label space $\mathcal{L}$.

4 Experiments

We experiment on the CoNLL 2005 (Carreras and Màrquez, 2005) and CoNLL 2012 (OntoNotes 5.0, (Pradhan et al., 2013)) benchmarks, using two SRL setups: *end-to-end* and *gold predicates*. In the *end-to-end* setup, a system takes a tokenized sentence as input, and predicts all the predicates and their arguments. Systems are evaluated on the micro-averaged F1 for correctly predicting (predicate, argument span, label) tuples. For comparison with previous systems, we also report results with *gold predicates*, in which the complete set of predicates in the input sentence is given as well. Other experimental setups and hyperparameteres are listed in Appendix A.1.

ELMo embeddings To further improve performance, we also add ELMo word representations (Peters et al., 2018) to the BiLSTM input (in the +ELMo rows). Since the contextualized representations ELMo provides can be applied to most previous neural systems, the improvement is orthogonal to our contribution. In Table 1 and 2, we organize all the results into two categories: the comparable single model systems, and the models augmented with ELMo or ensembling (in the PoE rows).

End-to-end results As shown in Table 1,[2] our joint model outperforms the previous best pipeline system (He et al., 2017) by an F1 difference of anywhere between 1.3 and 6.0 in every setting. The improvement is larger on the Brown test set, which is out-of-domain, and the CoNLL 2012 test set, which contains nominal predicates. On all datasets, our model is able to predict over 40% of the sentences completely correctly.

Results with gold predicates To compare with additional previous systems, we also conduct experiments with gold predicates by constraining our predicate beam to be gold predicates only. As shown in Table 2, our model significantly out-performs He et al. (2017), but falls short of Tan et al. (2018), a very recent attention-based (Vaswani et al., 2017) BIO-tagging model that was developed concurrently with our work. By adding the contextualized ELMo representations, we are able to out-perform all previous systems, including Peters et al. (2018), which applies ELMo to the SRL model introduced in He et al. (2017).

5 Analysis

Our model's architecture differs significantly from previous BIO systems in terms of both input and decision space. To better understand our model's strengths and weaknesses, we perform three analyses following Lee et al. (2017) and He et al. (2017), studying (1) the effectiveness of beam

[2]For the end-to-end setting on CoNLL 2012, we used a subset of the train/dev data from previous work due to noise in the dataset; the dev result is not directly comparable. See Appendix A.2 for detailed explanation.

	CoNLL 05 In-domain (WSJ)				Out-of-domain (Brown)			CoNLL 2012 (OntoNotes)			
End-to-End	Dev. F1	P	R	F1	P	R	F1	Dev. F1	P	R	F1
Ours+ELMo	**85.3**	**84.8**	**87.2**	**86.0**	**73.9**	**78.4**	**76.1**	**83.0**	**81.9**	**84.0**	**82.9**
He et al. (2017)$^{\text{PoE}}$	81.5	82.0	83.4	82.7	69.7	70.5	70.1	77.2	80.2	76.6	78.4
Ours	**81.6**	**81.2**	**83.9**	**82.5**	**69.7**	**71.9**	**70.8**	**79.4**	**79.4**	**80.1**	**79.8**
He et al. (2017)	80.3	80.2	82.3	81.2	67.6	69.6	68.5	75.5	78.6	75.1	76.8

Table 1: End-to-end SRL results for CoNLL 2005 and CoNLL 2012, compared to previous systems. CoNLL 05 contains two test sets: WSJ (in-domain) and Brown (out-of-domain).

	WSJ	Brown	OntoNotes
Ours+ELMo	**87.4**	**80.4**	**85.5**
Peters et al. (2018)+ELMo	-	-	84.6
Tan et al. (2018)$^{\text{PoE}}$	86.1	74.8	83.9
He et al. (2017)$^{\text{PoE}}$	84.6	73.6	83.4
FitzGerald et al. (2015)$^{\text{PoE}}$	80.3	72.2	80.1
Ours	83.9	73.7	82.1
Tan et al. (2018)	**84.8**	**74.1**	**82.7**
He et al. (2017)	83.1	72.1	81.7
Yang and Mitchell (2017)	81.9	72.0	-
Zhou and Xu (2015)	82.8	69.4	81.1

Table 2: Experiment results with gold predicates.

Arg. Beam	Φ_a	Pred. Beam	Φ_p
by ambulance	2.5	says	0.1
her mother ... ambulance	2.2	transported	0.0
her mother	2.2	ambulance	-8.3
Priscilla	1.9	been	-11.3
should	1.8		
transported by ambulance	-0.3		
Priscilla says .... ambulance	-2.2		
ambulance	-3.2		

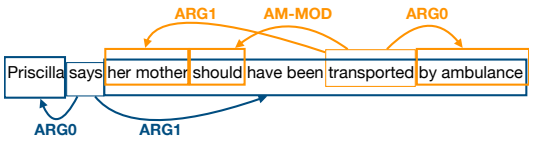

Figure 4: Top: The candidate arguments and predicates in the argument beam B_a and predicate beam B_p after pruning, along with their unary scores. Bottom: Predicted SRL relations with two identified predicates and their arguments.

pruning, (2) the ability to capture long-range dependencies, (3) agreement with syntactic spans, and (4) the ability to predict globally consistent SRL structures. The analyses are performed on the development sets without using ELMo embeddings. [3]

Effectiveness of beam pruning Figure 4 shows the predicate and argument spans kept in the beam, sorted with their unary scores. Our model efficiently prunes unlikely argument spans and predicates, significantly reduces the number of edges it needs to consider. Figure 5 shows the recall of predicate words on the CoNLL 2012 development set. By retaining $\lambda_p = 0.4$ predicates per word, we are able to keep over 99.7% argument-bearing predicates. Compared to having a part-of-speech tagger (POS:X in Figure 5), our joint beam pruning allowing the model to have a soft trade-off between efficiency and recall.[4]

Long-distance dependencies Figure 6 shows the performance breakdown by binned distance between arguments to the given predicates. Our model is better at accurately predicting arguments that are farther away from the predicates, even compared to an ensemble model (He et al., 2017) that has a higher overall F1. This is very likely due to architectural differences; in a BIO tagger, predicate information passes through many LSTM timesteps before reaching a long-distance argument, whereas our architecture enables direct connections between all predicates-arguments pairs.

Agreement with syntax As mentioned in He et al. (2017), their BIO-based SRL system has good agreement with gold syntactic span boundaries (94.3%) but falls short of previous syntax-based systems (Punyakanok et al., 2004). By directly modeling span information, our model achieves comparable syntactic agreement (95.0%) to Punyakanok et al. (2004) without explicitly modeling syntax.

Global consistency On the other hand, our model suffers from global consistency issues. For example, on the CoNLL 2005 test set, our model has lower complete-predicate accuracy (62.6%) than the BIO systems (He et al., 2017; Tan et al., 2018) (64.3%-66.4%). Table 3 shows its viola-

[3]For comparability with prior work, analyses (2)-(4) are performed on the CoNLL 05 dev set with gold predicates.

[4]The predicate ID accuracy of our model is not comparable with that reported in He et al. (2017), since our model does not predict non-argument-bearing predicates.

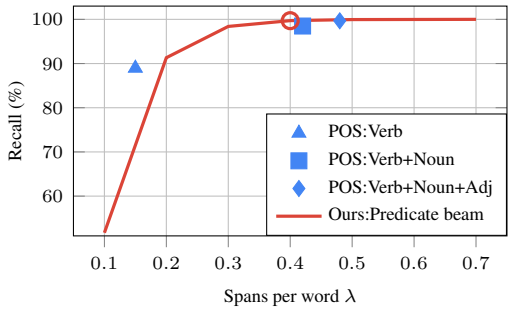

Figure 5: Recall of gold argument-bearing predicates on the CoNLL 2012 development data as we increase the number of predicates kept per word. POS:X shows the gold predicate recall from using certain pos-tags identified by the NLTK part-of-speech tagger (Bird, 2006).

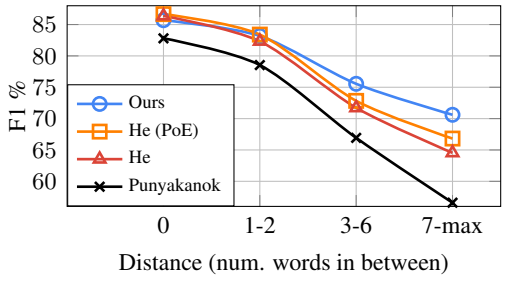

Figure 6: F1 by surface distance between predicates and arguments, showing degrading performance on long-range arguments.

tions of global structural constraints[5] compared to previous systems. Our model made more constraint violations compared to previous systems. For example, our model predicts duplicate core arguments[6] (shown in the U column in Table 3) more often than previous work. This is due to the fact that our model uses independent classifiers to label each predicate-argument pair, making it difficult for them to implicitly track the decisions made for several arguments with the same predicate.

The *Ours+decode* row in Table 3 shows SRL performance after enforcing the U-constraint using dynamic programming (Täckström et al., 2015) at decoding time. Constrained decoding at test time is effective at eliminating all the core-role inconsistencies (shown in the U-column), but did not bring significant gain on the end result (shown in SRL F1), which only evaluates the piece-wise predicate-argument structures.

[5] Punyakanok et al. (2008) described a list of global constraints for SRL systems, e.g., there can be at most one core argument of each type for each predicate.

[6] Arguments with labels ARG0,ARG1,...,ARG5 and AA.

			SRL-Violations		
Model/Oracle	SRL F1	Syn %	U	C	R
Gold	100.0	98.7	24	0	61
Ours+decode	82.4	95.1	0	8	104
Ours	82.3	95.0	69	7	105
He (PoE)	82.7	94.3	37	3	68
He	81.6	94.0	48	4	73
Punyakanok	77.4	95.3	0	0	0

Table 3: Comparison on the CoNLL 05 development set against previous systems in terms of unlabeled agreement with gold constituency (Syn%) and each type of SRL-constraints violations (**U**nique core roles, **C**ontinuation roles and **R**eference roles).

6 Conclusion and Future Work

We proposed a new SRL model that is able to jointly predict all predicates and argument spans, generalized from a recent coreference system (Lee et al., 2017). Compared to previous BIO systems, our new model supports joint predicate identification and is able to incorporate span-level features. Empirically, the model does better at long-range dependencies and agreement with syntactic boundaries, but is weaker at global consistency, due to our strong independence assumption.

In the future, we could incorporate higher-order inference methods (Lee et al., 2018) to relax this assumption. It would also be interesting to combine our span-based architecture with the self-attention layers (Tan et al., 2018; Strubell et al., 2018) for more effective contextualization.

Acknowledgments

This research was supported in part by the ARO (W911NF-16-1-0121), the NSF (IIS-1252835, IIS-1562364), a gift from Tencent, and an Allen Distinguished Investigator Award. We thank Eunsol Choi, Dipanjan Das, Nicholas Fitzgerald, Ariel Holtzman, Julian Michael, Noah Smith, Swabha Swayamdipta, and our anonymous reviewers for helpful feedback.

References

Steven Bird. 2006. Nltk: the natural language toolkit. In *ACL*.

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

Nicholas FitzGerald, Julian Michael, Luheng He, and Luke Zettlemoyer. 2018. Large-scale qa-srl parsing. In *ACL*.

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

Luheng He, Kenton Lee, Mike Lewis, and Luke S. Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *ACL*.

Lingpeng Kong, Chris Dyer, and Noah A Smith. 2016. Segmental recurrent neural networks. In *ICLR*.

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

Kenton Lee, Luheng He, and Luke Zettlemoyer. 2018. Higher-order coreference resolution with coarse-to-fine inference. In *NAACL*.

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

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *ACL*.

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. 2013. Towards robust linguistic analysis using ontonotes. In *CoNLL*.

Vasin Punyakanok, Dan Roth, Wen tau Yih, Dav Zimak, and Yuancheng Tu. 2004. Semantic role labeling via generalized inference over classifiers. In *CoNLL*.

Vasin Punyakanok, Dan Roth, and Wen-tau Yih. 2008. The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. A minimal span-based neural constituency parser. In *ACL*.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-Informed Self-Attention for Semantic Role Labeling. *arXiv preprint*.

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

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Bishan Yang and Tom M. Mitchell. 2017. A joint sequential and relational model for frame-semantic parsing. In *EMNLP*.

Yu Zhang, Guoguo Chen, Dong Yu, Kaisheng Yaco, Sanjeev Khudanpur, and James Glass. 2016. Highway long short-term memory rnns for distant speech recognition. In *ICASSP*.

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