How to best use Syntax in Semantic Role Labelling

Yufei Wang[1] and **Mark Johnson**[1] and **Stephen Wan**[2] and **Yifang Sun**[3] and **Wei Wang**[3]
Macquarie University, Sydney, Australia[1]
CSIRO Data61, Sydney, Australia[2]
The University of New South Wales, Sydney, Australia[3]
yufei.wang@students.mq.edu.au, mark.johnson@mq.edu.au
stephen.wan@data61.csiro.au
{yifangs,weiw}@cse.unsw.edu.au

Abstract

There are many different ways in which external information might be used in an NLP task. This paper investigates how external syntactic information can be used most effectively in the Semantic Role Labeling (SRL) task. We evaluate three different ways of encoding syntactic parses and three different ways of injecting them into a state-of-the-art neural ELMo-based SRL sequence labelling model. We show that using a constituency representation as input features improves performance the most, achieving a new state-of-the-art for non-ensemble SRL models on the in-domain **CoNLL'05** and **CoNLL'12** benchmarks.[1]

1 Introduction

Properly integrating external information into neural networks has received increasing attention recently (Wu et al., 2018; Li et al., 2017; Strubell et al., 2018). Previous research on this topic can be roughly categorized into three classes: **i) Input**: The external information are presented as additional input features (i.e., dense real-valued vectors) to the neural network (Collobert et al., 2011). **ii) Output**: The neural network is trained to predict the main task and the external information in a multi-task approach (Changpinyo et al., 2018). **iii) Auto-encoder**: This approach, recently proposed by Wu et al. (2018), simultaneously combines the **Input** and **Output** during neural models training. The simplicity of these methods allow them to apply to many NLP sequence tasks and various neural model architectures.

However, previous studies often focus on integrating word-level shallow features such as POS or chunk tags into the sequence labelling tasks. Syntactic information, which encodes the long-range dependencies and global sentence structure, has not been studied as carefully. This paper fills this gap by integrating syntactic information to the sequence labelling task. We address three questions: **1)** *How should syntactic information be encoded as word-level features?* **2)** *What is the best way of integrating syntactic information?* and **3)** *What effect does the choice of syntactic representation have on the performance?*

We study these questions in the context of Semantic Role Labelling (SRL). A SRL system extracts the predicate-argument structure of a sentence.[2] Syntax was an essential component of early SRL systems (Xue and Palmer, 2004; Punyakanok et al., 2008). The state-of-the-art neural SRL systems use a neural sequence labelling model without any syntax knowledge (He et al., 2018, 2017; Tan et al., 2018). We show below that injecting external syntactic knowledge into a neural SRL sequence labelling model can improve the performance, and our best model sets a new state-of-the-art for a non-ensemble SRL system.

In this paper we express the external syntactic information as vectors of discrete features, because this enables us to explore different ways of injecting the syntactic information into the neural SRL model. Specifically, we propose three different syntax encoding methods: **a)** a full constituency tree representation (**Full-C**); **b)** an SRL-specific span representation (**SRL-C**); and **c)** a dependency tree representation (**Dep**). For **(a)** we adapt the constituency parsing representation from (Gómez-Rodríguez and Vilares, 2018) and encode the tree structure as a set of features for word pairs. For **(b)**, we use a categorical representation of the constituency spans that are most relevant to SRL tasks based on (Xue and Palmer, 2004). Finally, **(c)** we propose a discrete vector representation that encodes the head-modifier relationships in the dependency trees.

We evaluate the effectiveness of these encodings using three different integration methods on

[1]Our model source code is available in https://github.com/GaryYufei/bestParseSRL

[2]*who* did *what* to *whom*, *where* and *when*

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 5338–5343
Florence, Italy, July 28 - August 2, 2019.

the SRL **CoNLL'05** and **CoNLL'12** benchmarks. We show that using either of the constituency representations in either the **Input** or the **Auto-Encoder** configurations produces the best performance. These results are noticeably better than a strong baseline and set a new state-of-the-art for non-ensemble SRL systems.

2 Related Work

Semantic Role Labeling (SRL) generally refers to the PropBank style of annotation (Palmer et al., 2005). Broadly speaking, prior work on SRL makes use of syntactic information in two different ways. Carreras and Màrquez (2005); Pradhan et al. (2013) incorporate constituent-structure span-based information, while Hajič et al. (2009) incorporate dependency-structure information.

This information can be incorporated into an SRL system in several different ways. Swayamdipta et al. (2018) use span information from constituency parse trees as an additional training target in a multi-task learning approach, similar to one of the approaches we evaluate here. Roth and Lapata (2016) use an LSTM model to represent the dependency paths between predicates and arguments and feed the output as the input features to their SRL system. Marcheggiani and Titov (2017) use Graph Convolutional Network (Niepert et al., 2016) to encode the dependency parsing trees into their LSTM-based SRL system. Xia et al. (2019) represent dependency parses using position-based categorical features of tree structures in a neural model. Strubell et al. (2018) use dependency trees as a supervision signal to train one of attention heads in a self-attentive neural model.

3 Syntactic Representation

This section introduces our representations of constituency and dependency syntax trees.

3.1 Full-C: Full Constituency Representation

Gómez-Rodríguez and Vilares (2018) propose a full representation of constituency parsing trees where the string position between w_i and w_{i+1} is associated with the pair $(n(w_i) - n(w_{i-1}), l(w_i))$ where $n(w_i)$ is the number of common ancestors between (w_i, w_{i+1}) and $l(w_i)$ is the non-terminal label at the lowest common ancestor[3]. For simplicity, we define $r(w_i) = n(w_i) - n(w_{i-1})$ throughout this paper. [4]

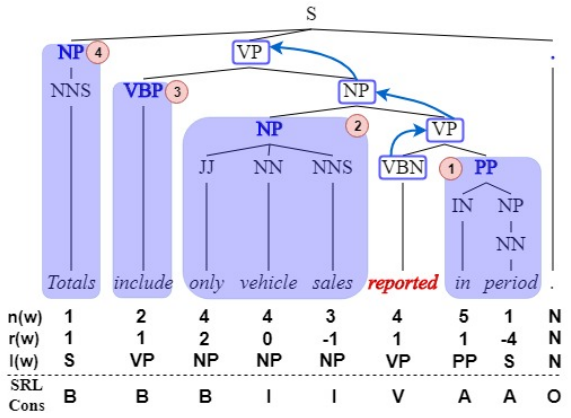


Figure 1: Examples of **Full-C** ($n(w)$, $r(w)$ and $l(w)$) and **SRL-C** (SRL-Cons). *reported* is the predicate word. The blue non-terminals are candidate constituents in the **SRL-C**. The circled number is the extraction order.

This encoding method transforms the whole constituency parsing tree into $n-1$ $(r(w_i), l(w_i))$ feature pairs for a length-n sentence. We assign $(r(w_i), l(w_i))$ to the w_i $(0 < i \leq n-1)$ and leave a padding symbol **N** to the w_n. We treat $r(w_i)$ and $l(w_i)$ as two separate categorical features for each word. We refer this representation as the **Full-C** (Figure 1).

3.2 SRL-C: SRL Span Representation

Xue and Palmer (2004) show only a small fraction of the constituents in the parse trees are useful for the SRL task given the predicate word. That means encoding the full constituency parsing tree may introduce redundant information.

Therefore, we preserve the constituent spans that are most likely to be useful for the predicate word in the trees. We re-use the *pruning algorithm* in (Xue and Palmer, 2004). Their algorithm collects the potential argument constituents by walking up the tree to the root node recursively, which filters out many irrelevant constituents from the syntax trees with 99.3% of the ground truth arguments preserved.

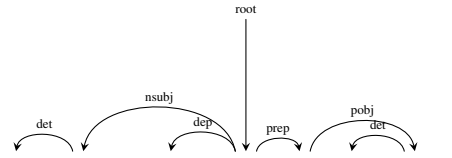
We encode the output of this rule-based pruning algorithm using a standard **BIO** (**B**egin-**I**nside-**O**utside) annotation scheme. The words that are

[3]The full constituency trees can be reconstructed from this representation, details refer to (Gómez-Rodríguez and Vilares, 2018)

[4]In (Gómez-Rodríguez and Vilares, 2018), both $r(w_i)$ and $n(w_i)$ is applicable for this encoding method. Our pilot experiments show that $r(w_i)$ works much better than the absolute representation $n(w_i)$.

outside any candidate constituent receive the tag **O**. The words that are beginning of a candidate constituent receive the tag **B**, and the words that are inside a candidate constituent receive the tag **I**. We use the tag **A** to label words in prepositional phrases. We refer this representation as the **SRL-C** (Figure 1).

3.3 Dep: Dependency Tree Representation



	The	seeds	already	are	in	the	script
Left	0	1	0	2	0	0	1
Right	0	0	0	1	1	0	0
RG	1	2	1	4	-1	1	-2
Edge	L	L	N	R	R	L	R
DL	det	nsubj	dep	root	prep	det	pobj

Figure 2: Features from Dependency Tree.

The dependency tree representation encodes key aspects of the head-modifier relationships within the sentence. We also consider encoding constituent edge information. The following word-level features have been proposed:

a) *#left/right Dependents* (**Left / Right**). The number of dependents a word has on the left and right side.

b) *Right/Left-most Dependent* (**Edge**). Whether the word is the Right/Left/None-most dependent of its governor.

c) *Relative Distance to Governor* (**RG**). The relative distance between the word and its governor.

d) *Dependency Label* (**DL**). The label describing the relationship between each pair of dependent and governor.

We refer this representation as the **Dep** (Figure 2[5]).

4 Injecting External Information

In this section, we introduce three different methods for integrating external syntactic information into the neural SRL system (Figure 3):

[5]In this example, we assume the "root" is the first word of the sentence from the left.

Figure 3: Model Architecture. Blue indicates the baseline model; Red indicates the multi-task output component; Green indicates the external feature component.

Baseline Our baseline system is a stacked bi-LSTM architecture (He et al., 2017). We use ELMo (Peters et al., 2018) as word embeddings and a CRF output decoder on the top of LSTM, as shown in Figure 3.

Input This approach represents the external categorical features as trainable, high dimensional dense vector token embeddings, which are concatenated with the representation vectors of ELMo in the baseline model. The syntactic parse trees that are used as the input features are produced by Kitaev and Klein (2018) (for constituency parsing). The dependency trees are produced by transforming the constituency trees using Stanford CoreNLP toolkit. This ensures that the constituency and dependency parses have a similar error distribution, helping to control for parsing quality. Our constituency and dependency parses score a state-of-the-art 95.4 F1 and 96.4% UAS on the WSJ test set respectively. We used a 20-fold cross-validation procedure to produce the data for the external syntactic input.

Output In this approach, our model predicts both SRL sequence tags and syntactic features (encoded as the word-level features above) simultaneously. We use a log loss for each categorical feature. The final training loss is the multi-task objective $L_{SRL} - \sum_{f=1}^{m} \log p_f(y_f^\star)$, where $p_f(y_f)$ is the probability of generating y_f as the f^{th} feature (m features in total, $m = 1, 2, 5$ for **SRL-C**, **Full-C** and **Dep** respectively) and $y_f^\star$ is the ground truth for the f^{th} feature. Gold training data was used as the external syntactic information for the multi-task output setting, as this external information is not required at test time.

Auto-encoder Following Wu et al. (2018), we use external information as input features and as a multi-task training objective simultaneously, so the system is behaving somewhat like an auto-encoder. This auto-encoder has to reproduce the syntactic information in its output that it is fed in

its input, encouraging it to incorporate this information in its internal representations. The input and output representations are the same as above.

5 Experiments

We evaluate 10 different models (the 3 ways of using external information by 3 different encodings of syntax and a baseline model) on **CoNLL'05** (Carreras and Màrquez, 2005) and **CoNLL'12** (Pradhan et al., 2013) benchmarks, under the evaluation setting where the gold predicate is given. The **CoNLL'05** benchmark uses WSJ and Brown test as in-domain and out-domain evaluation respectively.

5.1 Main Results

Table 1 shows the effect of using the three different kinds of external syntactic information in the three different ways just described. When used as input features, all three representations improve over our baseline system. This shows that syntactic representations provide additional useful information, which is beyond the dynamic context embeddings from ELMo, to SRL task.

Syntax Representations Models using constituency representations are 0.3% - 0.6% better than the models using the dependency representations. This might be because constituents align more directly with SRL arguments and constituency information is easier to use.

Inject.	Model	CoNLL'05		CoNLL'12
		WSJ	Brown	Test
-	Baseline	87.7	78.1	85.8
Input	**Full-C**	88.1	78.9	**86.4**
	SRL-C	**88.2**	**79.3**	**86.4**
	Dep	87.9	78.4	86.1
Output	**Full-C**	87.7	78.4	85.9
	SRL-C	87.9	78.5	85.9
	Dep	87.6	78.9	85.8
Auto Encoder	**Full-C**	**88.2**	77.7	86.3
	SRL-C	**88.2**	79.0	**86.4**
	Dep	87.6	78.1	85.7

Table 1: Injecting External Syntax Information. **Bold number** is the best performance in each column, same below.

The **SRL-C** is slightly better than the **Full-C** for in-domain evaluation. The advantages of the **SRL-C** approach are greater on the out-of-domain (Brown) evaluation, with a margin of 0.4%. This could be because **Full-C** is more sensitive to parsing errors than **SRL-C**. When we compare gold and automatic parser representations in Brown device data, 10.5% of the words get different **Full-C** features while this only 7.9% get different **SRL-C** features.

External Information Injection Table 1 shows at least on this task, multi-task learning does not perform as well as adding external information as additional input features. Both the *Input* and *Auto-Encoder* methods work equally well. We conclude that the extra complexity of the *auto-encoder* model is not justified. In particular, **Dep** with *auto-encoder* hurts SRL accuracy (0.6% behind the model with the constituency features).

5.2 Comparison with existing systems

We compare our best system (**SRL-C** used as Input) with previous work in Table 2. We improve upon the state-of-the-art results for non-ensemble SRL models on in-domain test by 0.6% and 0.2% on **CoNLL'05** and **CoNLL'12** respectively. Our model also achieves a competitive result on **CoNLL'05** Brown Test. Comparing with the strong ensemble model in (Ouchi et al., 2018), our model is only 0.3% and 0.6% lower in two benchmarks respectively.

Model	CoNLL'05		CoNLL'12
	WSJ	Brown	Test
ELMo Baseline	87.7	78.1	85.8
Strubell et al. (2018)	86.0	76.5	-
Xia et al. (2019)	86.9	76.8	-
He et al. (2018)	87.4	**80.4**	85.5
Ouchi et al. (2018)	87.6	78.7	86.2
Our best model	**88.2**	79.3	**86.4**
Xia et al. (2019)§	87.8	78.8	-
Ouchi et al. (2018)§	88.5	79.6	87.0

Table 2: Comparison with existing systems. § indicates ensemble models.

5.3 Using Gold Parse Trees

Finally, we conduct an oracle experiment where all syntactic features are derived from gold trees. Our model performance improves by around 3% - 4% F1 score (see Table 3). This bounds the improvement in SRL that one can expect with improved syntactic parses.

Model	CoNLL'05		CoNLL'12
	WSJ	Brown	Test
Our best model	88.2	79.3	86.4
Full-C	**92.2**	**83.5**	**91.4**
SRL-C	91.7	83.4	90.3
Dep	91.9	83.3	91.1

Table 3: SRL Performance with Gold Trees

6 Conclusion and Future Work

This paper evaluated three different ways of representing external syntactic parses, and three different ways of injecting that information into a state-of-the-art SRL system. We showed that representing the external syntactic information as constituents was most effective. Using the external syntactic information as input features was far more effective than a multi-task learning approach, and just as effective as an auto-encoder approach. Our best system sets a new state-of-the-art for non-ensemble SRL systems on in-domain data.

In future work we will explore how external information is best used in ensembles of models for SRL and other tasks. For example, is it better for all the models in an ensemble to use the same external information, or is it more effective if they make use of different kinds of information? We will also investigate whether the choice of method for injecting external information has the same impact on other NLP tasks as it does on SRL.

Acknowledgments

This research was supported by the Australian Research Councils Discovery Projects funding scheme (project number DPs 160102156, 170103710, 180103411), D2DCRC (DC25002, DC25003), and in part by CSIRO Data61.

References

Xavier Carreras and Lluís Màrquez. 2005. Introduction to the conll-2005 shared task: Semantic role labeling. In *Proceedings of the Ninth Conference on Computational Natural Language Learning (CoNLL-2005)*, pages 152–164. Association for Computational Linguistics.

Soravit Changpinyo, Hexiang Hu, and Fei Sha. 2018. Multi-task learning for sequence tagging: An empirical study. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2965–2977. Association for Computational Linguistics.

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

Carlos Gómez-Rodríguez and David Vilares. 2018. Constituent parsing as sequence labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1314–1324. Association for Computational Linguistics.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL 2009): Shared Task*, pages 1–18, Boulder, Colorado. Association for Computational Linguistics.

Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. 2018. Jointly predicting predicates and arguments in neural semantic role labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 364–369. Association for Computational Linguistics.

Luheng He, Kenton Lee, Mike Lewis, and Luke Zettlemoyer. 2017. Deep semantic role labeling: What works and what's next. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 473–483. Association for Computational Linguistics.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2676–2686. Association for Computational Linguistics.

Junhui Li, Deyi Xiong, Zhaopeng Tu, Muhua Zhu, Min Zhang, and Guodong Zhou. 2017. Modeling source syntax for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 688–697. Association for Computational Linguistics.

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*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. 2016. Learning convolutional neural networks for graphs. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2014–2023. JMLR.org.

Hiroki Ouchi, Hiroyuki Shindo, and Yuji Matsumoto. 2018. A span selection model for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1630–1642. Association for Computational Linguistics.

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

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using ontonotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152. Association for Computational Linguistics.

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. 2016. Neural semantic role labeling with dependency path embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1192–1202, Berlin, Germany. Association for Computational Linguistics.

Emma Strubell, Patrick Verga, Daniel Andor, David Weiss, and Andrew McCallum. 2018. Linguistically-informed self-attention for semantic role labeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5027–5038. Association for Computational Linguistics.

Swabha Swayamdipta, Sam Thomson, Kenton Lee, Luke Zettlemoyer, Chris Dyer, and Noah A. Smith. 2018. Syntactic scaffolds for semantic structures. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3772–3782. Association for Computational Linguistics.

Zhixing Tan, Mingxuan Wang, Jun Xie, Yidong Chen, and Xiaodong Shi. 2018. Deep semantic role labeling with self-attention. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4929–4936.

Minghao Wu, Fei Liu, and Trevor Cohn. 2018. Evaluating the utility of hand-crafted features in sequence labelling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2850–2856. Association for Computational Linguistics.

Qingrong Xia, Zhenghua Li, Min Zhang, Meishan Zhang, Guohong Fu, Rui Wang, and Luo Si. 2019. Syntax-aware neural semantic role labeling. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, (AAAI-19), Honolulu, Hawaii, USA, Jan 27-Feb 1, 2019*.

Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 88–94.