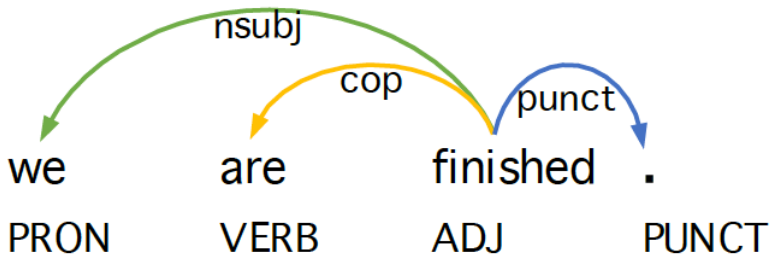
Joint models for POS tagging and dependency parsing, and for NER and relation classification

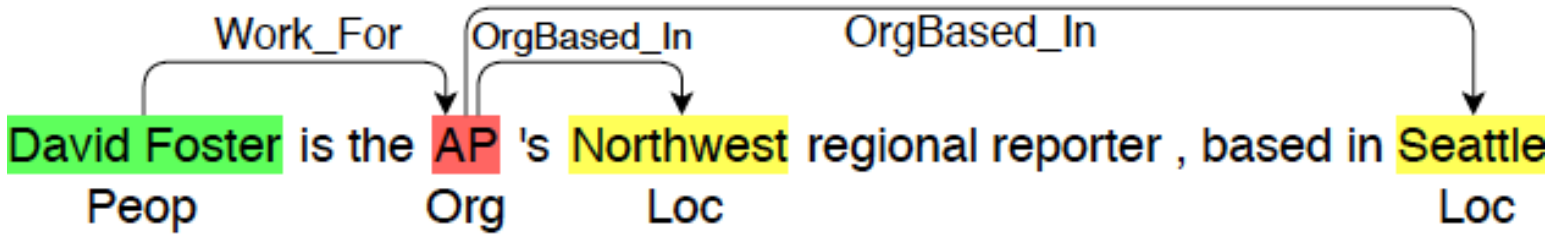
Dat Quoc Nguyen

joint work with

Karin Verspoor

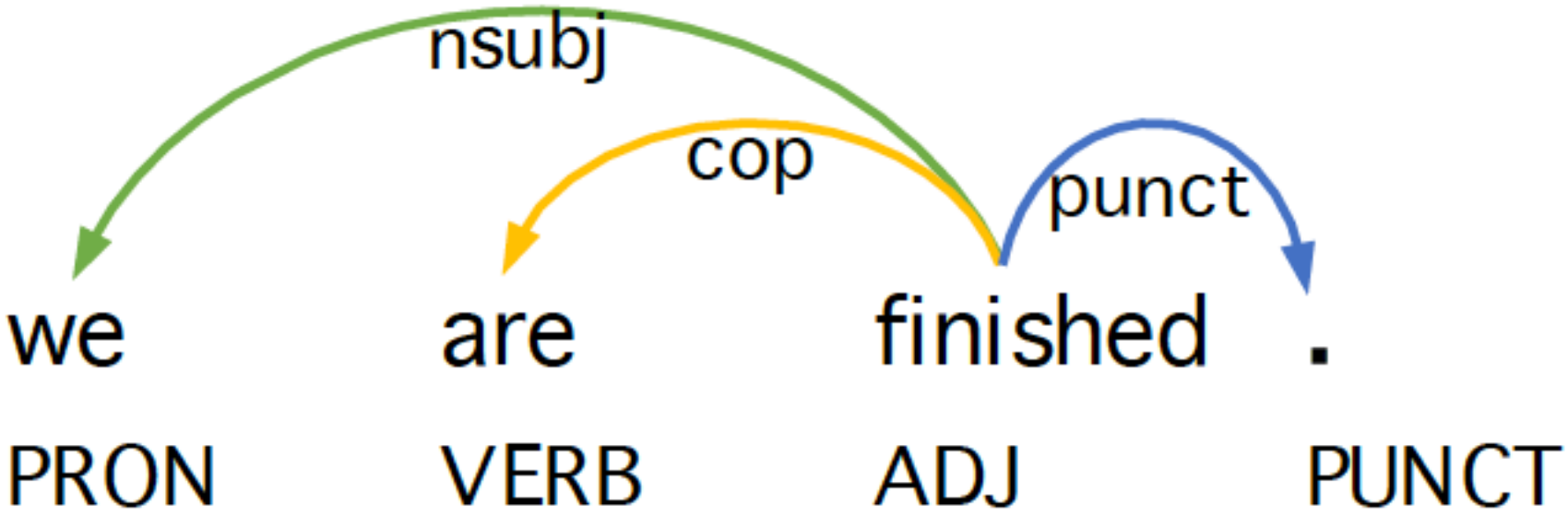
11/10/2018





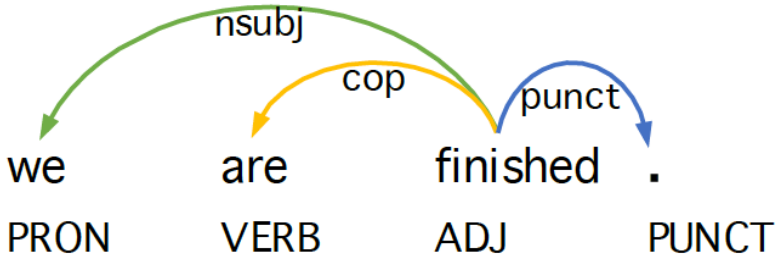
POS Tagging and Dependency Parsing

- cop: copula
- nsubj: nominal subject

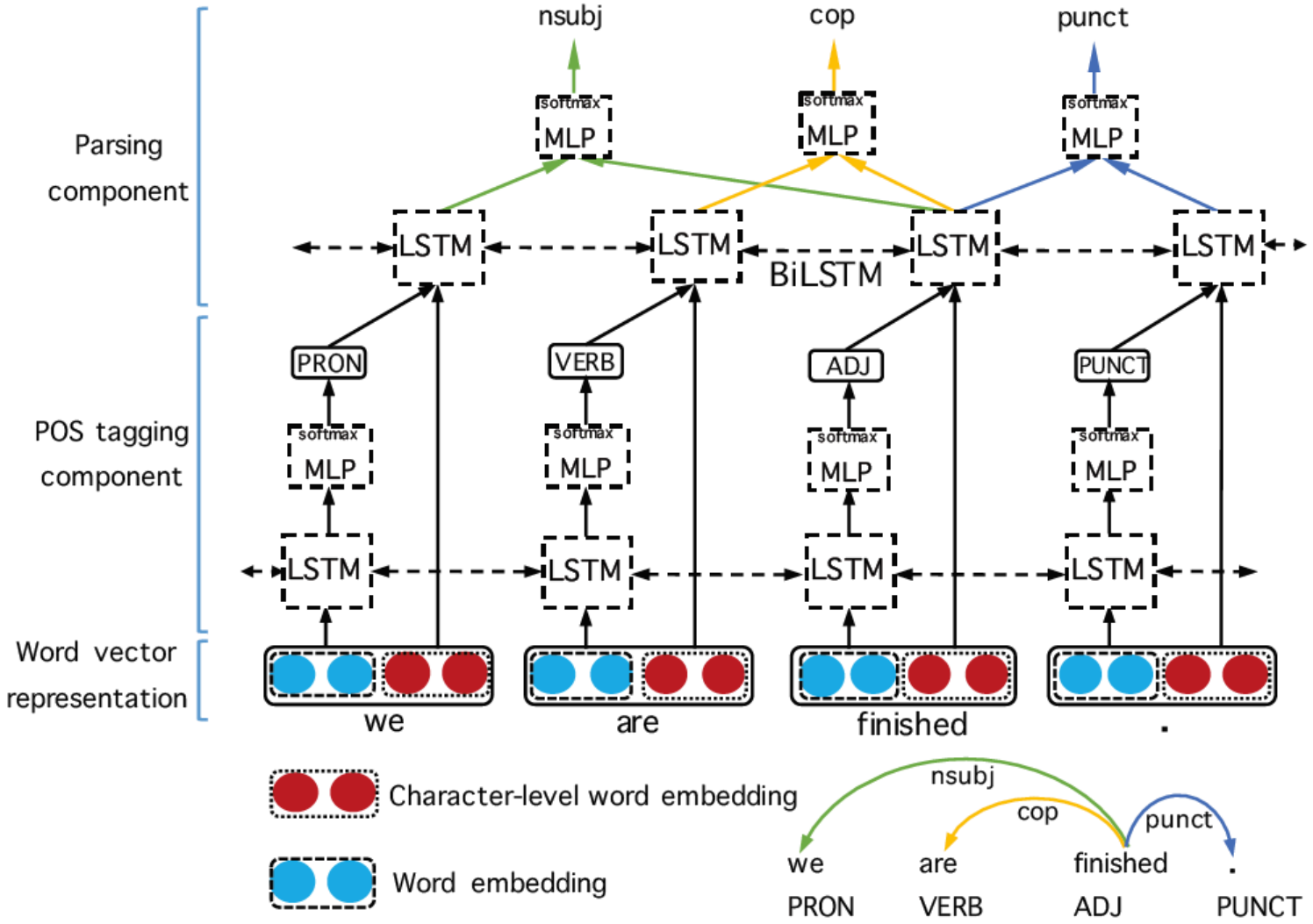


Joint POS Tagging and Dependency Parsing

- In real-world parsing, dependency parsers rely heavily on the use of automatically predicted POS tags, thus encountering error propagation problems:
  - Li et al. (2011) and Straka et al. (2016) show that parsing accuracies drop by 5+% when using automatic POS tags instead of gold ones
- Joint learning both POS tagging and dependency parsing:
  - More accurate POS tags could lead to improved parsing performance, and
  - The syntactic context of a parse tree could help resolve POS ambiguities

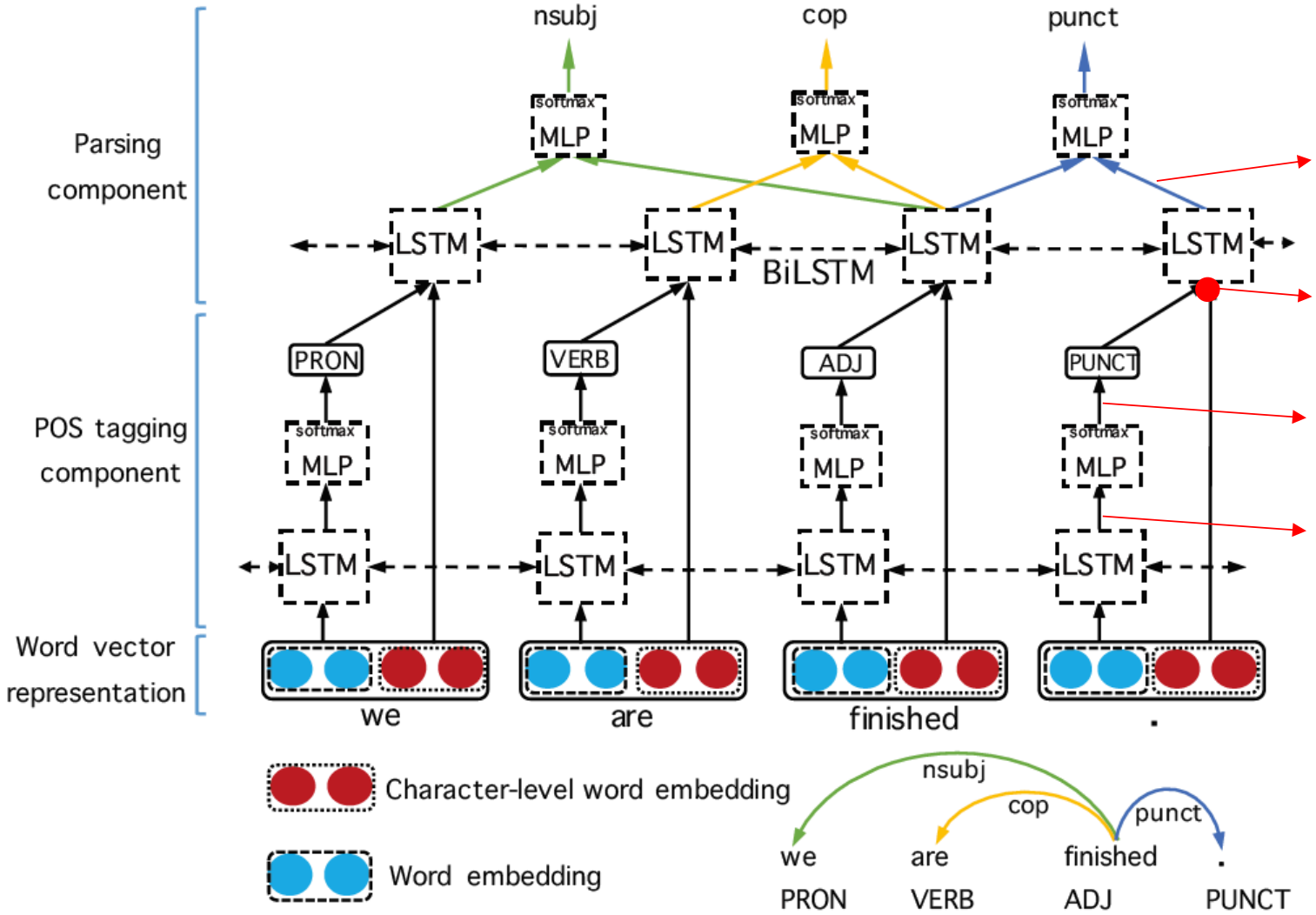


Joint POS Tagging and Dependency Parsing



- A two-component mixture of a tagging component and a parsing component
- The tagging component uses a standard BiLSTM to learn “latent” feature vectors for POS tag prediction
- The parsing component uses another BiLSTM to learn a set of latent features, then feeds these features into a MLP to decode dependency arcs and another MLP to label the predicted dependency arcs

Joint POS Tagging and Dependency Parsing



$$\boldsymbol{v}_i = \mathrm{BiLSTM}_{\mathrm{dep}}(\mathbf{x}_{1:n}, i)$$

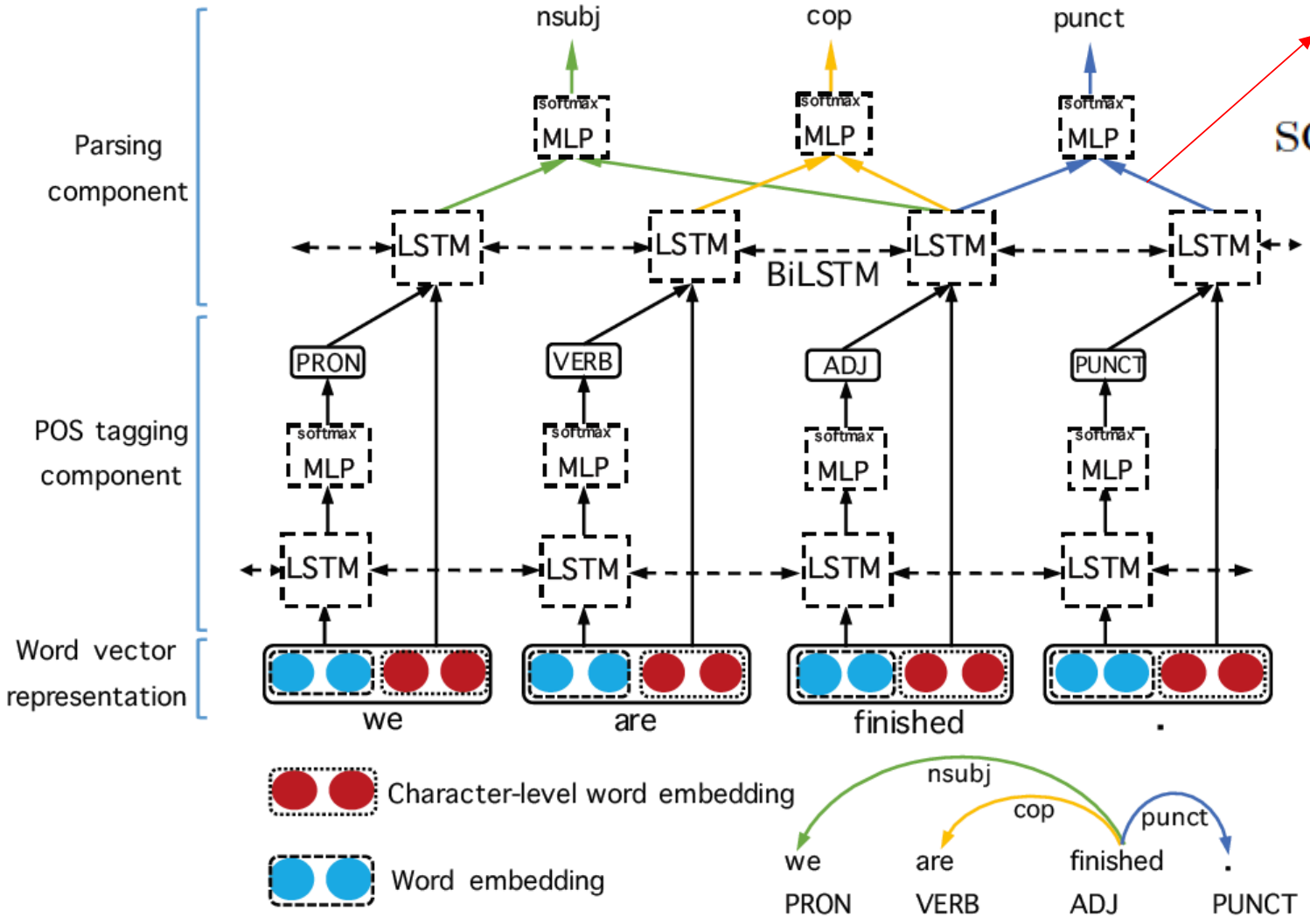
$$\mathbf{x}_i = \mathbf{e}^{(\mathrm{P})}_{p_i} \circ \mathbf{e}_i$$

$$\boldsymbol{\vartheta}_i = \mathrm{MLP}_{\mathrm{pos}}(\boldsymbol{v}^{(\mathrm{pos})}_i)$$

$$\boldsymbol{v}^{(\mathrm{pos})}_i = \mathrm{BiLSTM}_{\mathrm{pos}}(\mathbf{e}_{1:n}, i)$$

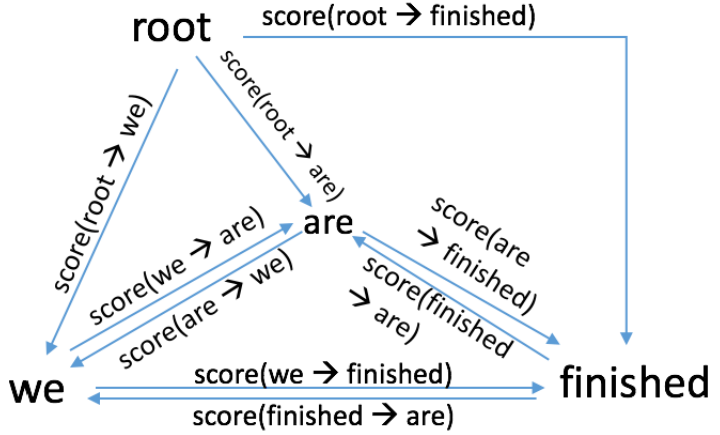
$$\mathbf{e}_i = \mathbf{e}^{(\mathrm{w})}_{w_i} \circ \mathbf{e}^{(\mathrm{c})}_{w_i}$$

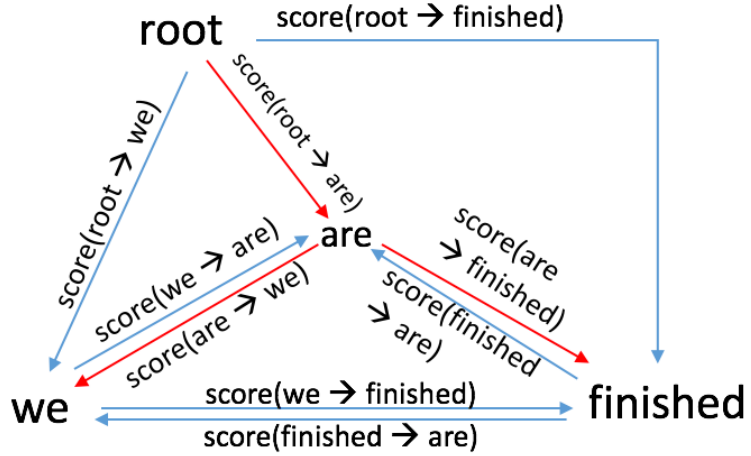
Joint POS Tagging and Dependency Parsing



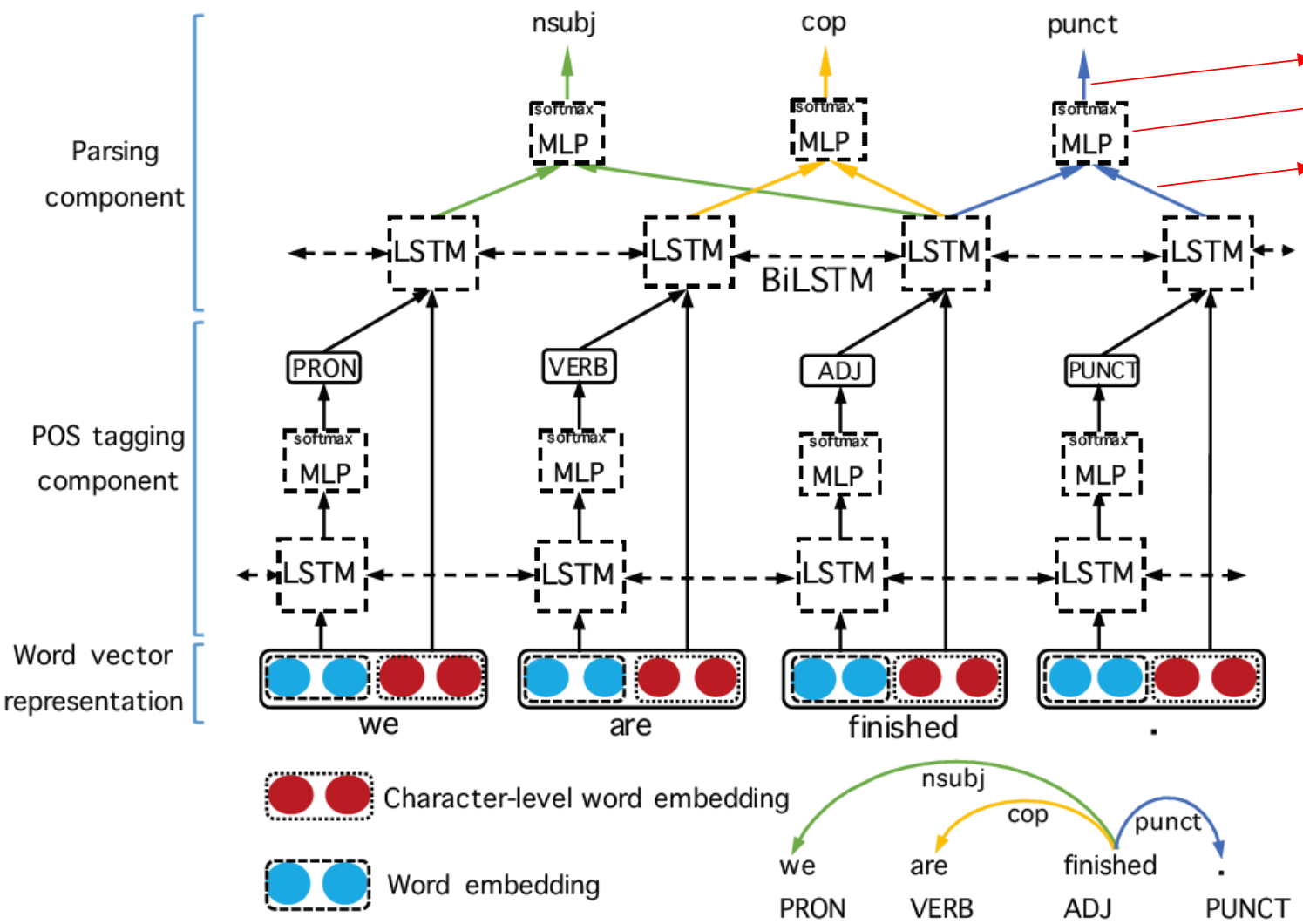
$$\boldsymbol{v}_i = \mathrm{BiLSTM}_{\mathrm{dep}}(\mathbf{x}_{1:n}, i)$$

$$\mathrm{score}_{\mathrm{arc}}(i, j) = \mathrm{MLP}_{\mathrm{arc}}(\boldsymbol{v}_i \circ \boldsymbol{v}_j)$$





Joint POS Tagging and Dependency Parsing



$$\mathbf{v}_{(h,m)} = \mathrm{MLP}_{\mathrm{rel}}\left(\boldsymbol{v}_h \circ \boldsymbol{v}_m\right)$$

$$\boldsymbol{v}_i = \mathrm{BiLSTM}_{\mathrm{dep}}\left(\mathbf{x}_{1:n}, i\right)$$

- Joint training:

$$\mathcal{L} = \mathcal{L}_{\mathrm{POS}} + \mathcal{L}_{\mathrm{ARC}} + \mathcal{L}_{\mathrm{REL}}$$

- POS tagging cross-entropy loss
- Arc-structure loss: margin between the gold unlabeled parse tree and the highest scoring incorrect tree
- Relation labeling cross-entropy loss

Joint POS Tagging and Dependency Parsing

- Penn WSJ treebank results

(2 BiLSTM layers)

#states	With punctuations			Without pun.	
	POS	UAS	LAS	UAS	LAS
128	**97.64**	93.68	92.11	94.42	92.61
256	97.63	**93.89**	**92.33**	**94.63**	**92.82**
Chen and Manning (2014)				92.0	89.7
Dyer et al. (2015)				93.2	90.9
BIST-graph [K&G16]				93.3	91.0
Zhang et al. (2017)				94.30	91.95
Ma and Hovy (2017)				94.77	92.66
Dozat and Manning (2017)				**95.24**	**93.37**

Table 1: Results on the development set. **#states**

Model	POS	UAS	LAS
Chen and Manning (2014)	97.3	91.8	89.6
Dyer et al. (2015)	97.3	93.1	90.9
Weiss et al. (2015)	97.44	93.99	92.05
BIST-graph [K&G16]	97.3	93.1	91.0
BIST-transition [K&G16]	97.3	93.9	91.9
Kuncoro et al. (2016)	97.3	94.26	92.06
Andor et al. (2016)	97.44	94.61	92.79
Zhang et al. (2017)	97.3	94.10	91.90
Ma and Hovy (2017)	97.3	94.88	92.98
Dozat and Manning (2017)	97.3	**95.44**	**93.76**
Dozat and Manning (2017) [•]	97.3	95.66	94.03
Bohnet and Nivre (2012) [⋆]	97.42	93.67	92.68
Alberti et al. (2015)	97.44	94.23	92.36
Zhang and Weiss (2016)	_	93.43	91.41
Hashimoto et al. (2017)	_	**94.67**	**92.90**
Yang et al. (2018)	97.54	94.18	92.26
Our model	**97.97**	94.51	92.87

Table 2: Results on the test set. POS tagging accu-

Joint POS Tagging and Dependency Parsing

- The CoNLL 2018 shared task on UD parsing
  - Fixed set of hyper-parameters (2 BiLSTM layers and 128-dimensional LSTM hidden states)

	System	All (82)	Big (61)	PUD (5)	Small (7)	Low (9)
UPOS	UDPipe 1.2	87.32	93.71	85.23	**87.36**	45.20
	UniMelb	**87.90**	**94.50**	**85.33**	87.12	45.20
UAS	UDPipe 1.2	71.64	78.78	71.22	63.17	30.08
	UniMelb	**74.16**	**81.83**	**73.17**	**64.71**	30.08
LAS	UDPipe 1.2	65.80	74.14	66.63	55.01	17.17
	UniMelb	**68.65**	**77.69**	**68.72**	**56.12**	17.17

Joint POS Tagging and Dependency Parsing

Results: 2017 shared task systems applied to 2018 test data

http://universaldependencies.org/conll18/results-2017-systems.html

LAS ranking 2017 systems

1.	Stanford (Stanford)	77.44
2.	C2L2 (Ithaca) ensemble	75.59
3.	ParisNLP (Paris)	72.67
4.	HIT-SCIR (Harbin)	71.67
5.	Koç University (İstanbul)	70.84
6.	TurkuNLP (Turku)	70.23
7.	ÚFAL - UDPipe 1.2 (Praha)	69.62
8.	MQuni (Sydney)	69.53
9.	UParse (Edinburgh)	69.44
10.	fbaml (Palo Alto)	69.34
11.	LyS-FASTPARSE (A Coruña)	68.76
12.	darc (Tübingen)	68.71

LAS ranking 2018 systems

1.	HIT-SCIR (Harbin)	82.23
2.	TurkuNLP (Turku)	79.67
3.	ICS PAS (Warszawa)	79.61
4.	UDPipe Future (Praha)	79.54
5.	Stanford (Stanford)	79.53
6.	LATTICE (Paris)	78.84
7.	CEA LIST (Paris)	78.76
8.	Uppsala (Uppsala)	78.24
9.	NLP-Cube (București)	78.03
10.	ParisNLP (Paris)	77.60
11.	AntNLP (Shanghai)	77.37
12.	SLT-Interactions (Bengaluru)	77.08
13.	UniMelb (Melbourne)	75.26

Use Stanford Biaffine and/or Improve Pre-process steps

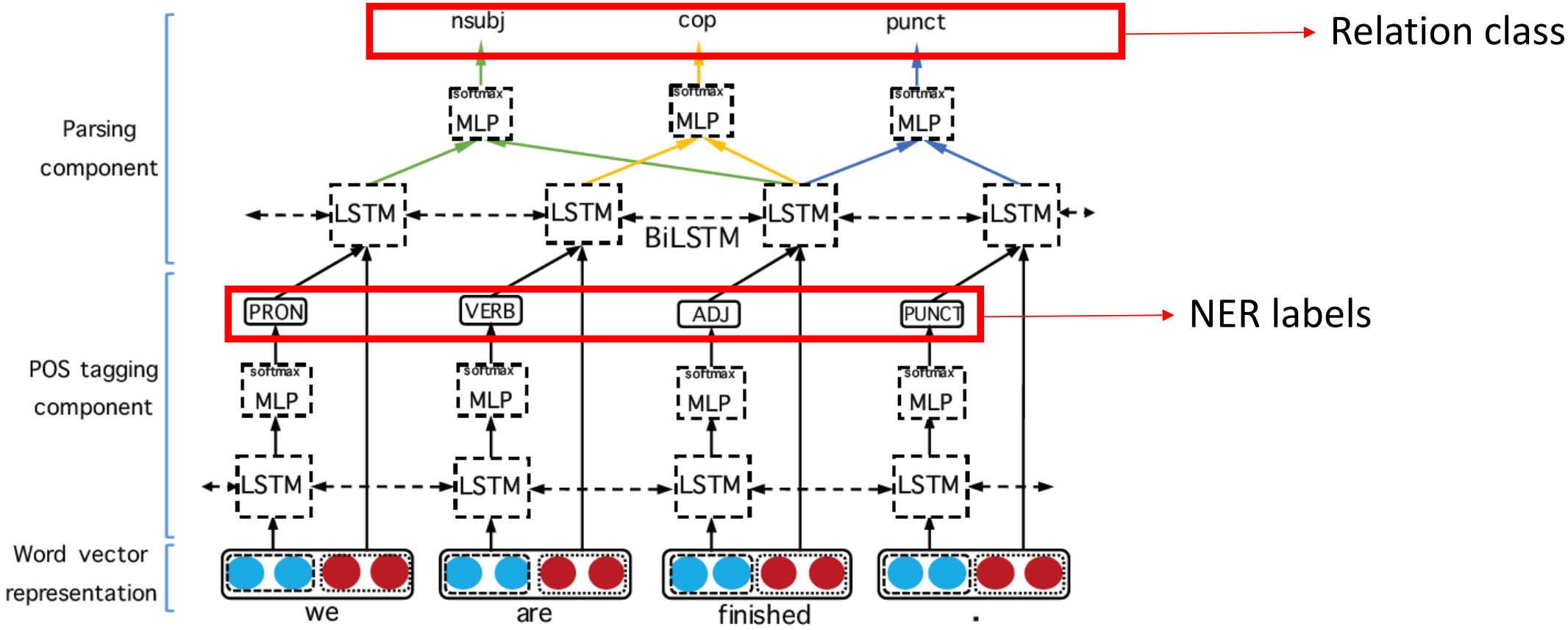
Joint POS Tagging and Dependency Parsing

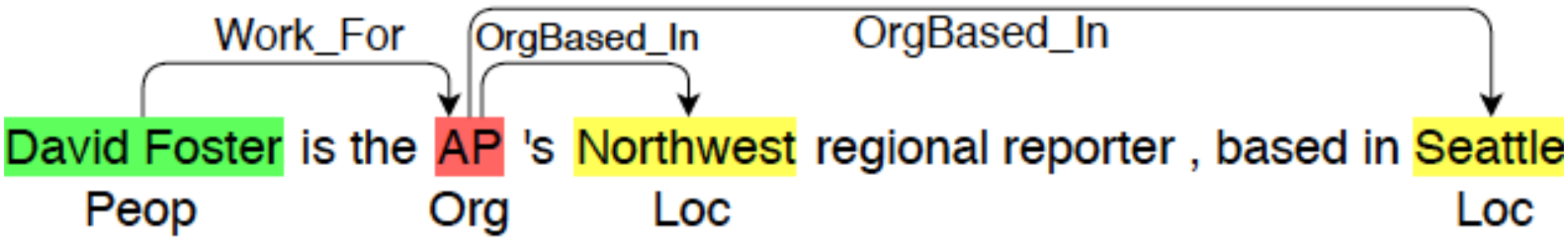
- The 2018 Extrinsic Parser Evaluation (EPE) campaign
  - Fixed set of hyper-parameters as used for the CoNLL 2018 shared task

Task	Development set				Evaluation set			
	Pre.	Rec.	F1	SP17	Pre.	Rec.	F1	SP17
Event extraction	57.87	51.20	**54.33**$_{1}$	52.67$_{54.59}$	58.52	49.43	**53.59**$_{1}$	50.29$_{50.23}$
Negation resolution	100	44.51	61.60$_{3}$	**64.85**$_{65.37}$	100	41.83	58.99$_{3}$	**65.13**$_{66.16}$
Opinion analysis	69.12	64.65	**66.81**$_{1}$	66.63$_{68.53}$	66.67	62.88	**64.72**$_{1}$	63.72$_{65.14}$
Average	_	_	60.91$_{1}$	**61.38**$_{62.83}$	_	_	59.10$_{1}$	**59.71**$_{60.51}$

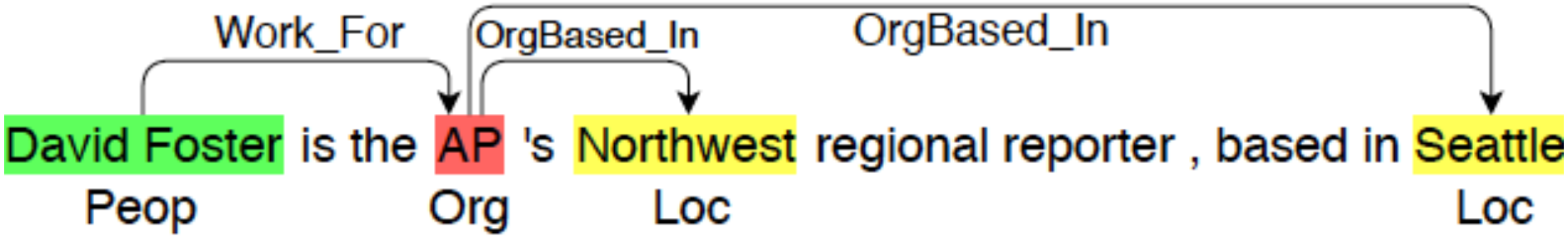
Table 5: Downstream task scores Precision (Prec.), Recall (Rec.) and F1 for our UniMelb team. The *subscript* in the F1 column denotes the unofficial rank of UniMelb over 17 participating teams at EPE 2018 (Fares et al., 2018). "SP17" denotes the F1 scores obtained by the EPE 2017 system Stanford-Paris (Schuster et al., 2017) with respect to (w.r.t.) the Stanford basic dependencies. The *subscript* in the SP17 column denotes the F1 scores obtained by Stanford-Paris w.r.t. the UD-v1-enhanced type of dependency representations, in which the average F1 score at 60.51 is the highest one at EPE 2017.

Joint POS Tagging and Dependency Parsing



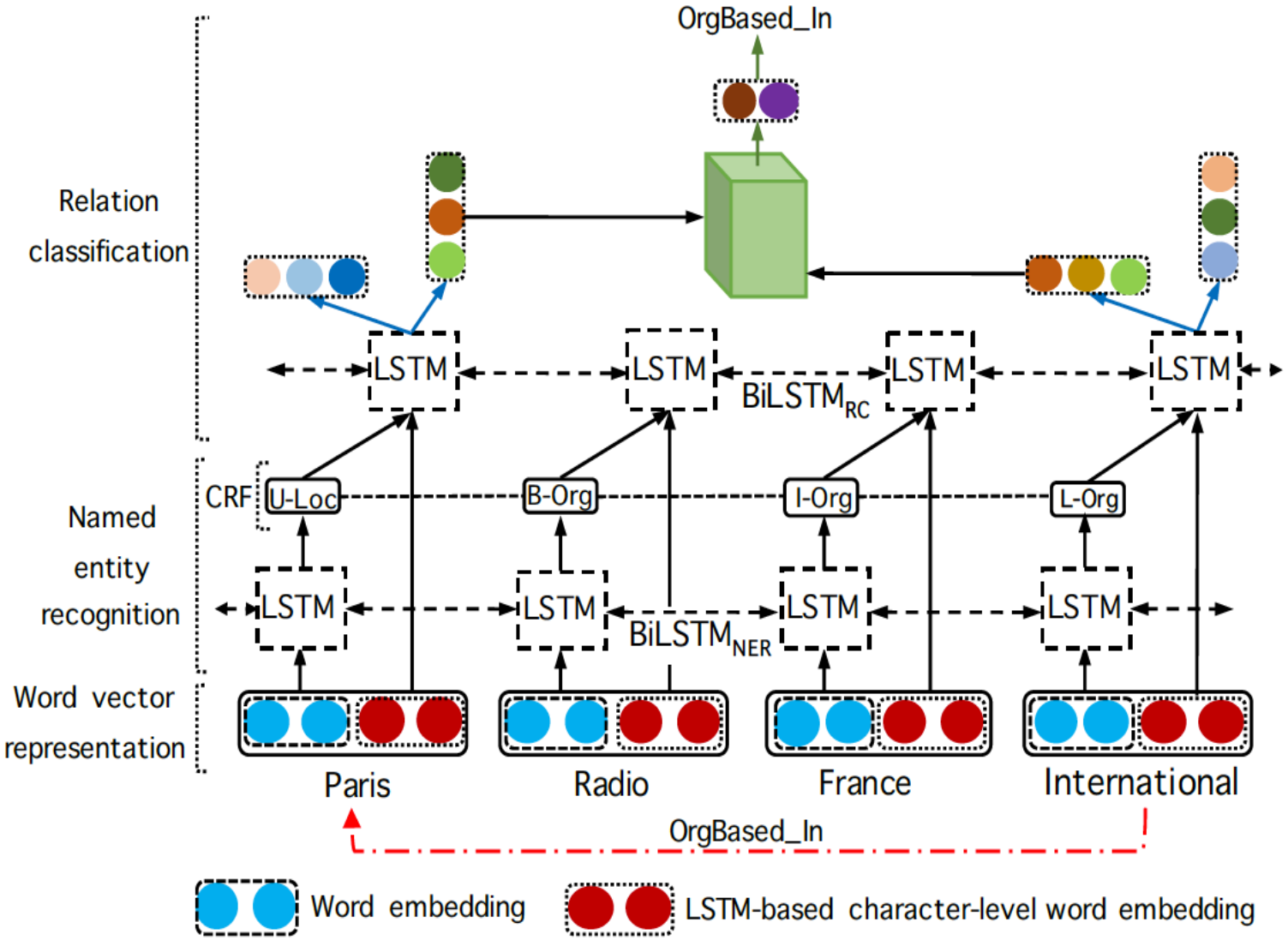


Relation extraction



- Extracting entities and their semantic relations from raw text is a key information extraction task
  - Such information is useful in many other NLP tasks
  - In IR applications such as entity search, structured search and question answering, it helps provide end users with significantly better search experience
- Recently, end-to-end systems which jointly learn to extract entities and relations have been proposed with strong potential to obtain high performance

End-to-end relation extraction for Joint NER and relation classification



Our NER component employs a standard BiLSTM-CRF architecture to predict entities from input word tokens

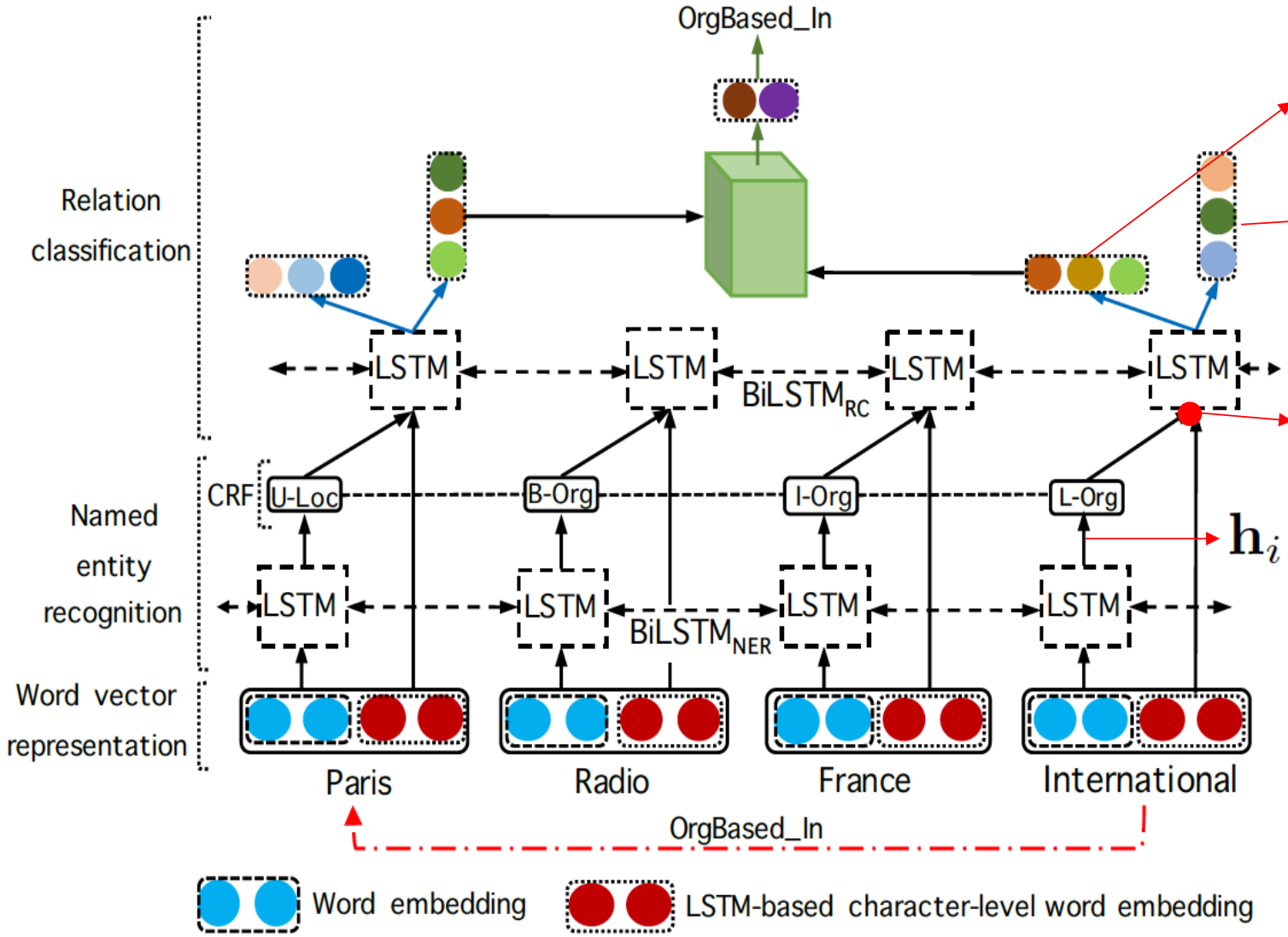
The RC component uses another BiLSTM to learn latent features relevant for relation classification

- For relation classification, we propose a novel use of the *biaffine* attention mechanism (Dozat and Manning, 2017) which was first introduced in dependency parsing

End-to-end relation extraction

$$\mathbf{s}_{j,k} = \text{Biaffine}\left(\mathbf{h}_j^{(\text{head})}, \mathbf{h}_k^{(\text{tail})}\right)$$

$$\text{Biaffine}(\mathbf{y}_1, \mathbf{y}_2) = \underbrace{\mathbf{y}_1^{\mathsf{T}}\mathbf{U}\mathbf{y}_2}_{\text{Bilinear}} + \underbrace{\mathbf{W}(\mathbf{y}_1 \circ \mathbf{y}_2) + \mathbf{b}}_{\text{Linear}}$$



$$\mathbf{h}_i^{(\text{head})} = \text{FFNN}_{\text{head}}(\boldsymbol{r}_i)$$

$$\mathbf{h}_i^{(\text{tail})} = \text{FFNN}_{\text{tail}}(\boldsymbol{r}_i)$$

$$\mathbf{r}_i = \text{BiLSTM}_{\text{RC}}(\boldsymbol{x}_{1:n}, i)$$

$$\mathbf{x}_i = \mathbf{e}_{t_i} \circ \mathbf{v}_i$$

$$\mathbf{h}_i = \text{FFNN}_{\text{NER}}\left(\text{BiLSTM}_{\text{NER}}(\mathbf{v}_{1:n}, i)\right)$$

$$\mathbf{v}_i = \mathbf{e}_{w_i}^{(\text{W})} \circ \mathbf{e}_{w_i}^{(\text{C})}$$

End-to-end relation extraction

Table 1. Comparison with the previous state-of-the-art results on the test set. Recall that Setup 2 uses gold entity boundaries while Setup 1 does not. The subscript denotes the standard deviation. **(F)** refers to the use of extra feature types such as POS tag-based or dependency parsing-based features. Although using the same test set, Gupta et al. (2016) [9] reported results on a 80/0/20 training/development/test split rather than our 64/16/20 split. Results in the last two rows are just for reference, not for comparison, due to a random sampling of the test set. In particular, Miwa and Sasaki (2014) [19] used the 80/0/20 split for Setup 1 and performed 5-fold cross validation (i.e. sort of equivalent to 80/0/20) for Setup 2, while Zhang et al. (2017) [30] used a 72/8/20 split.

Model	**Setup 1**		**Setup 2**	
	NER	RC	EC	RC
Gupta et al. (2016) [9]	_	_	88.8	58.3
Gupta et al. (2016) [9] **(F)**	_	_	92.4	69.9
Adel and Schütze (2017) [1]	_	_	82.1	62.5
Bekoulis et al. (2018) [4]	83.6	62.0	93.0	68.0
Bekoulis et al. (2018) [5]	83.9	62.0	93.3	67.0
Our joint model	**86.2**$_{0.5}$	**64.4**$_{0.6}$	**93.8**$_{0.4}$	**69.6**$_{0.7}$
Miwa and Sasaki (2014) [19] **(F)**	80.7	61.0	92.3	71.0
Zhang et al. (2017) [30] **(F)**	85.6	67.8	_	_

End-to-end relation extraction

Table 2. Ablation results on the development set. * and ** denote the statistically significant differences against the full results at $p < 0.05$ and $p < 0.01$, respectively (using the two-tailed paired t-test). (a) Without using the character-level word embeddings. (b) Using a softmax layer for NER label prediction instead of the CRF layer. (c) Without using the NER label embeddings in our RC component, i.e. Equation 3 would become $\mathbf{x}_i = \mathbf{v}_i$. (d) Without using the Bilinear part in Equation 8, i.e., Biaffine would be a common Linear mechanism. (e) Without using the Linear part in Equation 8, i.e., Biaffine reduces to Bilinear.

Model	**Setup 1**		**Setup 2**	
	NER	RC	EC	RC
Pipeline	$87.3_{0.6}$	$66.3_{0.8}$	$93.4_{0.6}$	$72.9_{0.6}$
Joint model (full)	$87.1_{0.5}$	$66.9_{0.8}$	$93.3_{0.5}$	$73.3_{0.6}$
(a) w/o Character	$82.7^{**}_{0.5}$	$63.0^{**}_{0.7}$	$93.1_{0.6}$	$73.4_{0.8}$
(b) w/o CRF	$86.4^{*}_{0.5}$	$66.0^{*}_{0.8}$	$93.5_{0.4}$	$73.2_{0.6}$
(c) w/o Entity	$87.1_{0.5}$	$64.7^{**}_{0.9}$	$93.3_{0.6}$	$72.1^{**}_{0.7}$
(d) w/o Bilinear	$86.6_{0.5}$	$65.4^{**}_{0.7}$	$93.4_{0.5}$	$72.0^{**}_{0.7}$
(e) w/o Linear	$86.8_{0.6}$	$65.9^{*}_{0.7}$	$93.3_{0.5}$	$72.6^{*}_{0.5}$

Thanks for your attention!

1. Dat Quoc Nguyen and Karin Verspoor. 2018. An improved neural network model for joint POS tagging and dependency parsing. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, to appear.
2. Dat Quoc Nguyen and Karin Verspoor. End-to-end neural relation extraction using deep biaffine attention. *Submitted to ECIR 2019*.