Do we need recursive subtree composition in dependency parsing?

Miryam de Lhoneux
@mdlhx

UPPSALA
UNIVERSITET

10 December 2019
Workshop on Data-driven Approaches to Parsing and Semantic Composition

Overview

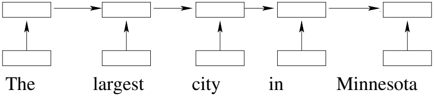
1. Tree vs. sequential LSTMs for parsing
2. BiLSTM parsing
3. Composition in a BiLSTM-parser
4. Composition for Auxiliary Verb Constructions
5. Conclusion

Outline for section 1

1. Tree vs. sequential LSTMs for parsing
2. BiLSTM parsing
3. Composition in a BiLSTM-parser
4. Composition for Auxiliary Verb Constructions
5. Conclusion

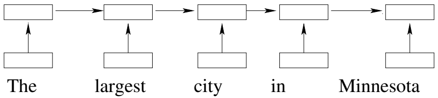
Recursive vs recurrent NNs

Recurrent

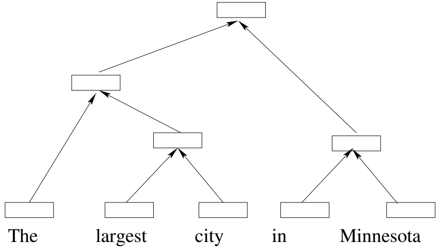


Recursive vs recurrent NNs

Recurrent



Recursive



Recursive NN for Transition-Based Parsing

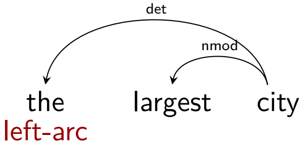
the largest city

Recursive NN for Transition-Based Parsing

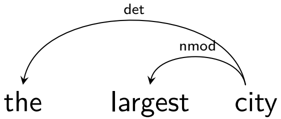
$$\text{the} \quad \text{largest} \xleftarrow{\text{nmod}} \text{city}$$

left-arc

Recursive NN for Transition-Based Parsing

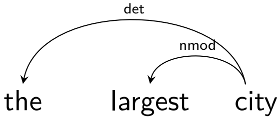


Recursive NN for Transition-Based Parsing



Recursive composition function in the stack-LSTM parser (Dyer et al., 2015):

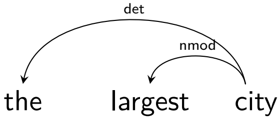
Recursive NN for Transition-Based Parsing



Recursive composition function in the stack-LSTM parser (Dyer et al., 2015):

$$c(h, d, r) = tanh(W[h; d; r] + b)$$

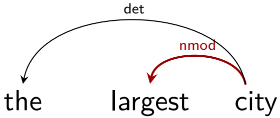
Recursive NN for Transition-Based Parsing



Recursive composition function in the stack-LSTM parser (Dyer et al., 2015):

$$c(h, d, r) = tanh(W[h; d; r] + b)$$
$$h_i = c(h_{i-1}, d, r)$$

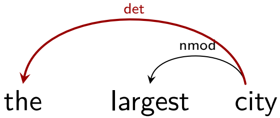
Recursive NN for Transition-Based Parsing



Recursive composition function in the stack-LSTM parser (Dyer et al., 2015):

$$\begin{aligned} c(h, d, r) &= tanh(W[h; d; r] + b) \\ city_1 &= c(city_0, largest, left - nmod) \end{aligned}$$

Recursive NN for Transition-Based Parsing



Recursive composition function in the stack-LSTM parser (Dyer et al., 2015):

$$c(h, d, r) = tanh(W[h; d; r] + b)$$

$$city_1 = c(city_0, largest, left - nmod)$$

$$city_2 = c(city_1, the, left - det)$$

Recursive vs recurrent NNs

Recursive vs recurrent NNs

	English PTB	Chinese CTB

Recursive vs recurrent NNs

	English PTB	Chinese CTB
S-LSTM without composition	89.6	83.6

Recursive vs recurrent NNs

	English PTB	Chinese CTB
S-LSTM without composition	89.6	83.6
S-LSTM with composition	90.9	85.7

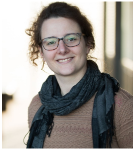
Recursive vs recurrent NNs

	English PTB	Chinese CTB
S-LSTM without composition	89.6	83.6
S-LSTM with composition	90.9	85.7
BiLSTM	91.2	85.0

Goals

Recursive Subtree Composition in LSTM-Based Dependency Parsing

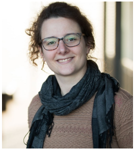
Miryam de Lhoneux♠ **Miguel Ballesteros**♢ **Joakim Nivre**♠
♠ Department of Linguistics and Philology, Uppsala University
♢ IBM Research AI, Yorktown Heights, NY



Goals

Recursive Subtree Composition in LSTM-Based Dependency Parsing

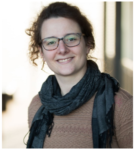
Miryam de Lhoneux[♠] **Miguel Ballesteros**[♢] **Joakim Nivre**[♠]
[♠] Department of Linguistics and Philology, Uppsala University
[♢] IBM Research AI, Yorktown Heights, NY



Goals

Recursive Subtree Composition in LSTM-Based Dependency Parsing

Miryam de Lhoneux♠ **Miguel Ballesteros**♢ **Joakim Nivre**♠
♠ Department of Linguistics and Philology, Uppsala University
♢ IBM Research AI, Yorktown Heights, NY

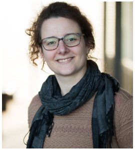


- BiLSTM + composition?

Goals

Recursive Subtree Composition in LSTM-Based Dependency Parsing

Miryam de Lhoneux♠ **Miguel Ballesteros**◊ **Joakim Nivre**♠
♠ Department of Linguistics and Philology, Uppsala University
◊ IBM Research AI, Yorktown Heights, NY

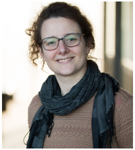
  

- BiLSTM + composition?
- Examine composition in simple architecture

Goals

Recursive Subtree Composition in LSTM-Based Dependency Parsing

Miryam de Lhoneux[♠] **Miguel Ballesteros**[♢] **Joakim Nivre**[♠]
[♠] Department of Linguistics and Philology, Uppsala University
[♢] IBM Research AI, Yorktown Heights, NY

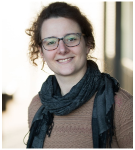
  

- BiLSTM + composition?
- Examine composition in simple architecture
- Typologically diverse languages

Goals

What Should/Do/Can LSTMs Learn When Parsing Auxiliary Verb Constructions?

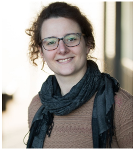
Miryam de Lhoneux, Sara Stymne and Joakim Nivre



Goals

What Should/Do/Can LSTMs Learn
When Parsing Auxiliary Verb Constructions?

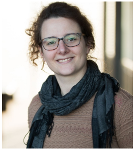
Miryam de Lhoneux, Sara Stymne and Joakim Nivre

Goals

What Should/Do/Can LSTMs Learn When Parsing Auxiliary Verb Constructions?

Miryam de Lhoneux, Sara Stymne and Joakim Nivre

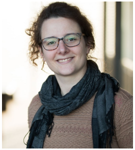


- Characterise what our parser learns about language

Goals

What Should/Do/Can LSTMs Learn When Parsing Auxiliary Verb Constructions?

Miryam de Lhoneux, Sara Stymne and Joakim Nivre

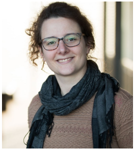
  

- Characterise what our parser learns about language
- Examine what our parser learns about auxiliary verb constructions (AVCs)

Goals

What Should/Do/Can LSTMs Learn When Parsing Auxiliary Verb Constructions?

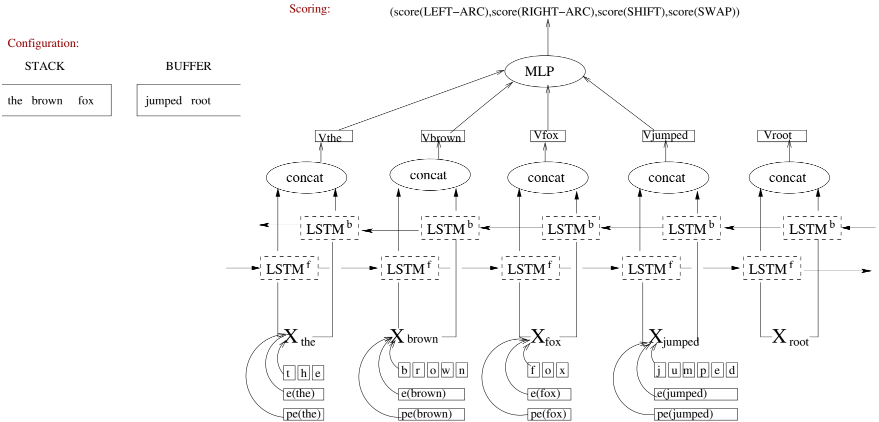
Miryam de Lhoneux, Sara Stymne and Joakim Nivre

- Characterise what our parser learns about language
- Examine what our parser learns about auxiliary verb constructions (AVCs)
- Investigate the role of composition for AVCs

Outline for section 2

Transition-Based Parsing using BiLSTMs

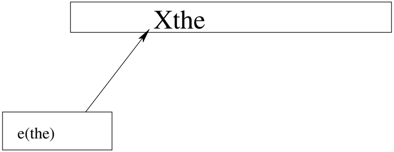


Kiperwasser and Goldberg (2016); de Lhoneux et al. (2017)

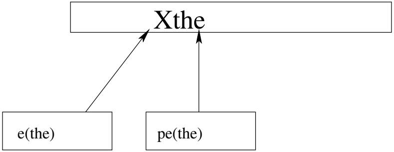
Transition-Based Parsing using BiLSTMs

Xthe

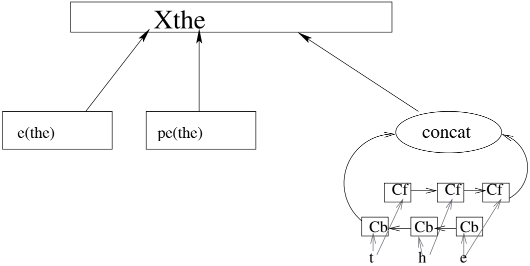
Transition-Based Parsing using BiLSTMs



Transition-Based Parsing using BiLSTMs



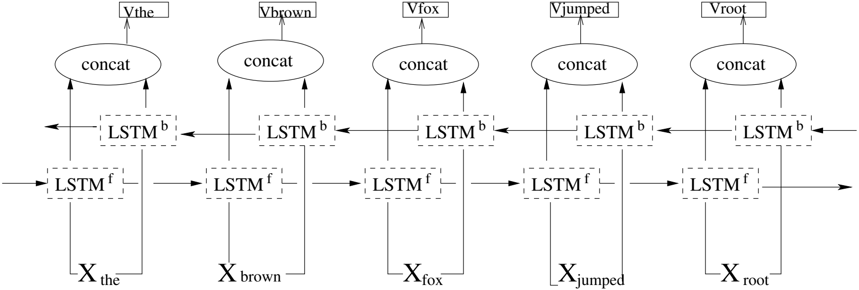
Transition-Based Parsing using BiLSTMs



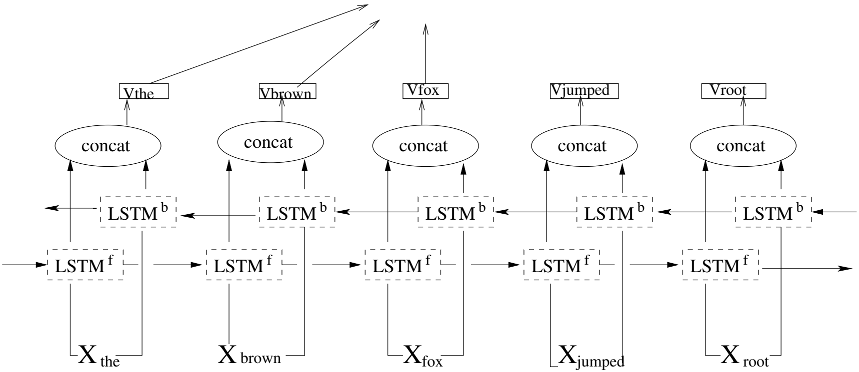
Transition-Based Parsing using BiLSTMs

X_{the} X_{brown} X_{fox} X_{jumped} X_{root}

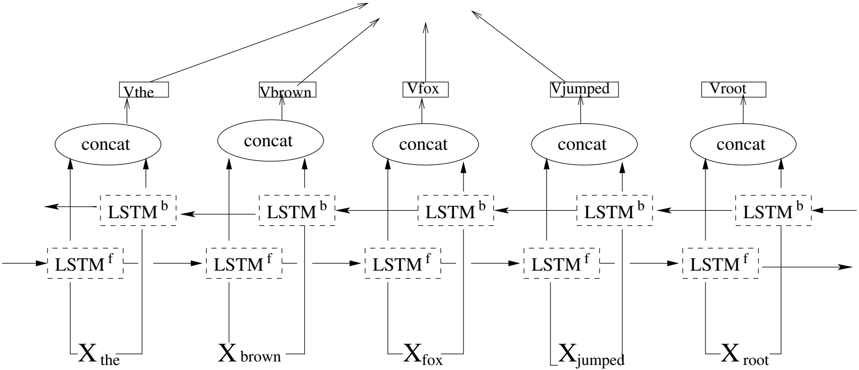
Transition-Based Parsing using BiLSTMs



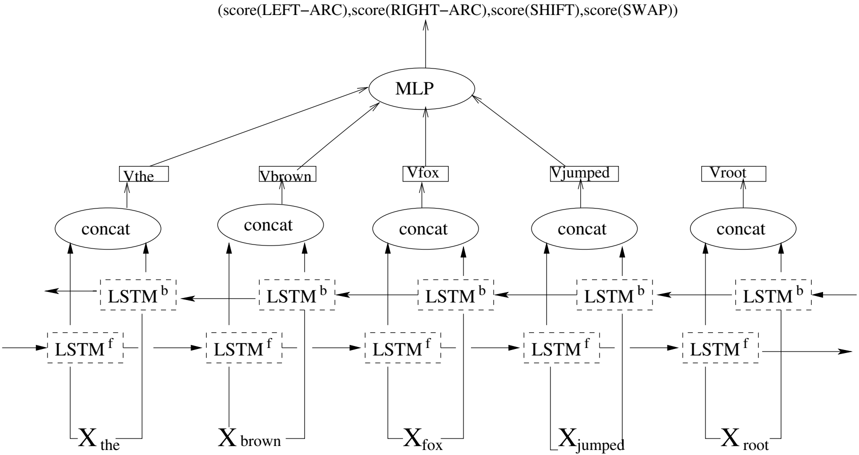
Transition-Based Parsing using BiLSTMs



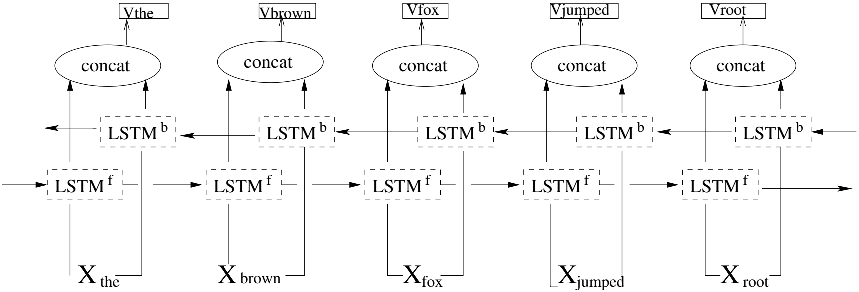
Transition-Based Parsing using BiLSTMs



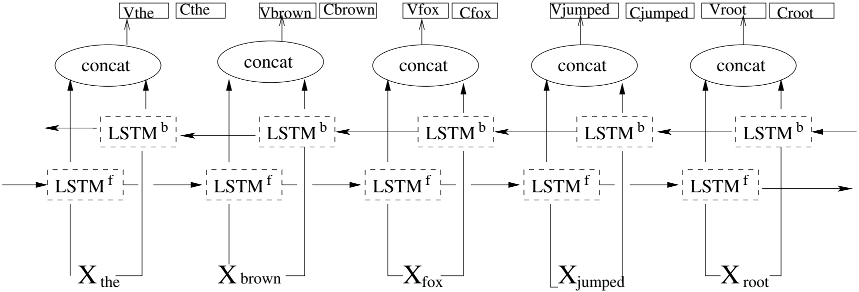
Transition-Based Parsing using BiLSTMs



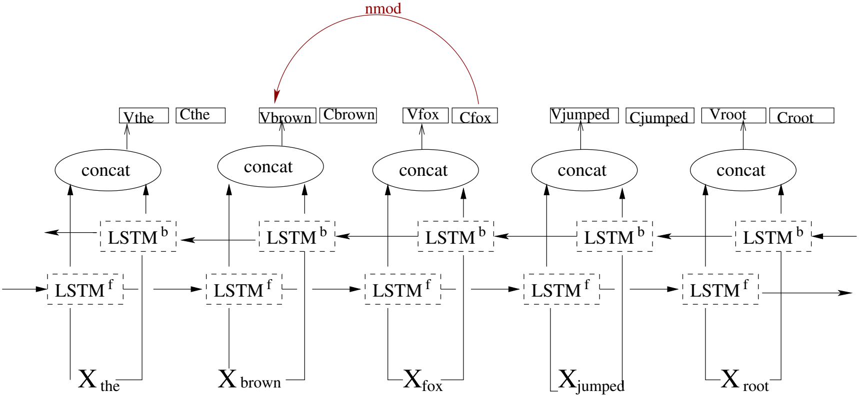
Recursive Composition in the BiLSTM parser



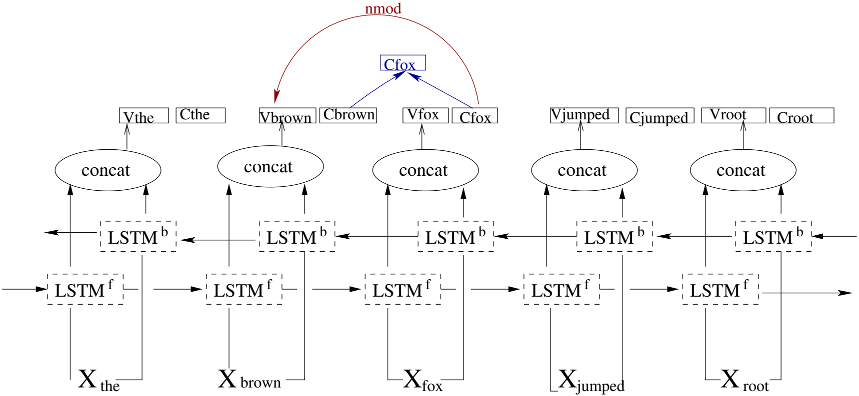
Recursive Composition in the BiLSTM parser



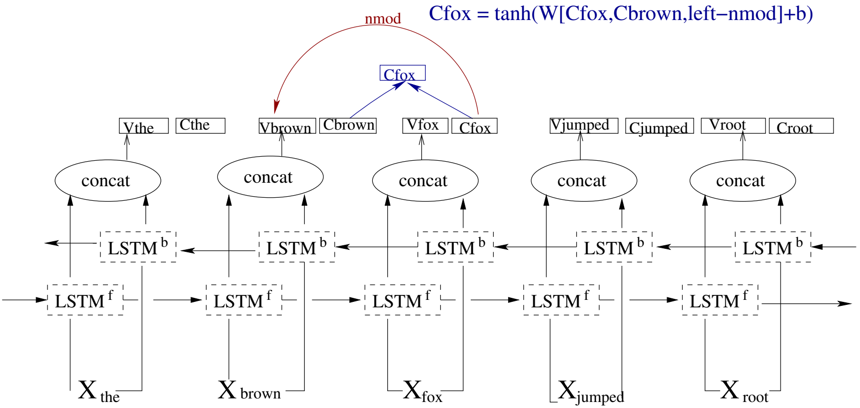
Recursive Composition in the BiLSTM parser



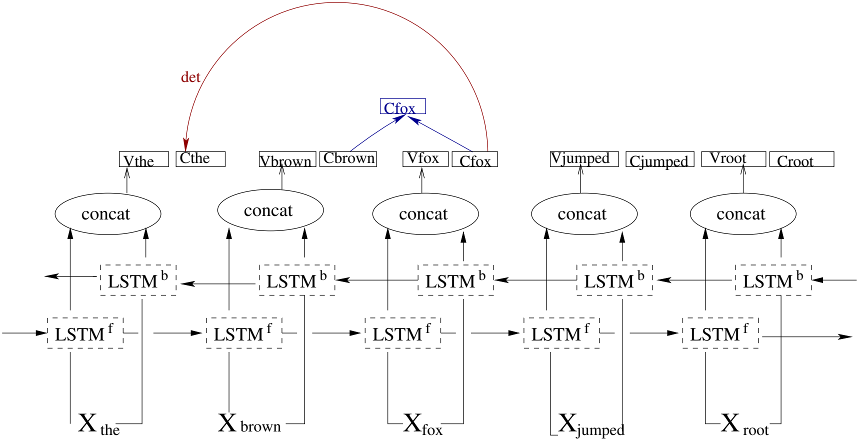
Recursive Composition in the BiLSTM parser



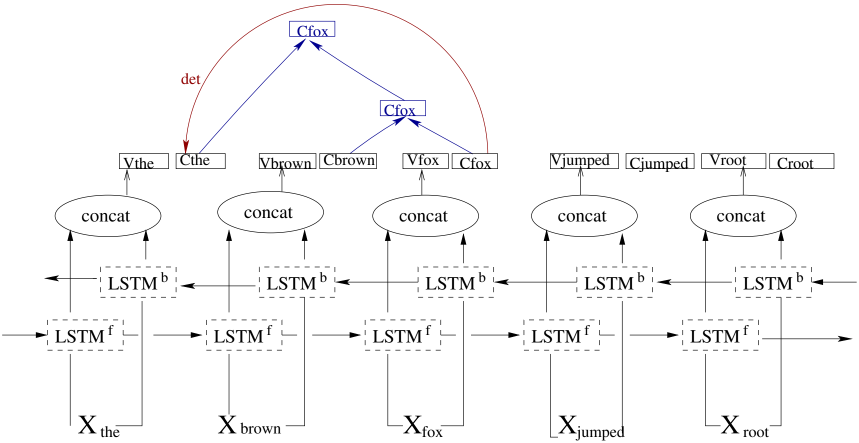
Recursive Composition in the BiLSTM parser



Recursive Composition in the BiLSTM parser



Recursive Composition in the BiLSTM parser



Recursive Composition in the BiLSTM parser

$$c_{head} = tanh(W[h; d; r] + b)$$

Recursive Composition in the BiLSTM parser

$$c_{head} = tanh(W[h; d; r] + b) + rc$$

Recursive Composition in the BiLSTM parser

$$c_{head} = tanh(W[h; d; r] + b) + rc$$

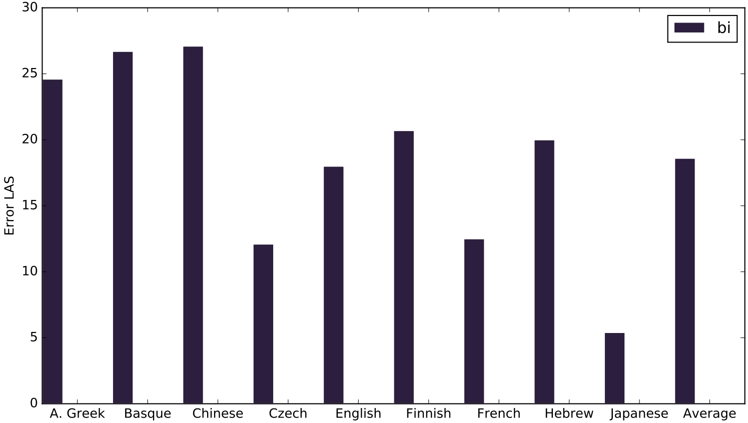
$$c_{head} = \textsc{Lstm}([h; d; r])$$

Recursive Composition in the BiLSTM parser

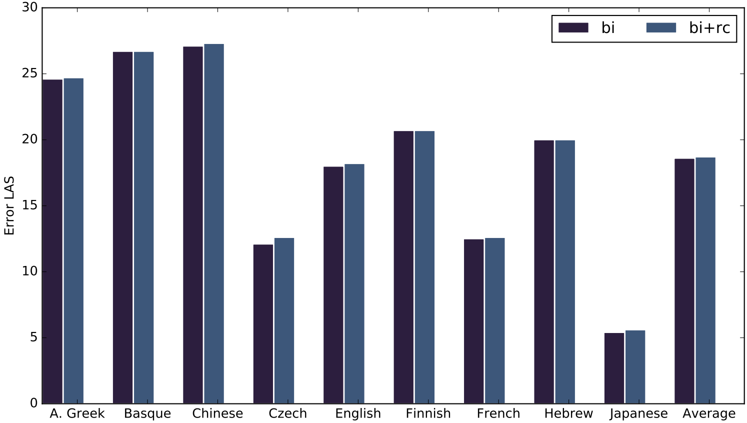
$$c_{head} = tanh(W[h; d; r] + b) + rc$$
$$c_{head} = \textsc{Lstm}([h; d; r]) + lc$$

Outline for section 3

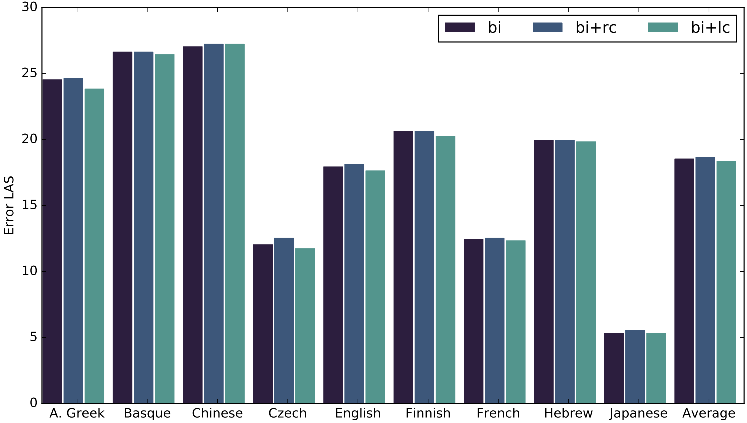
Results: BiLSTM + composition



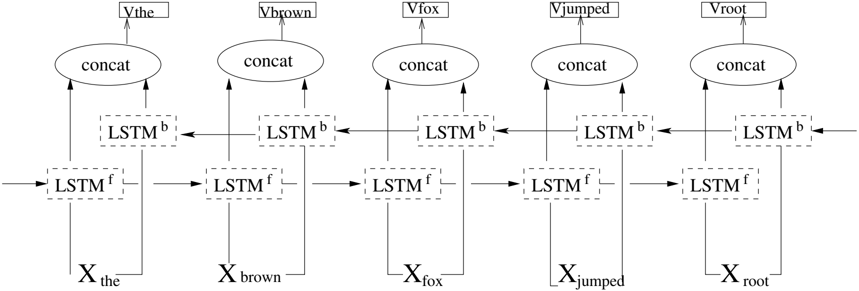
Results: BiLSTM + composition



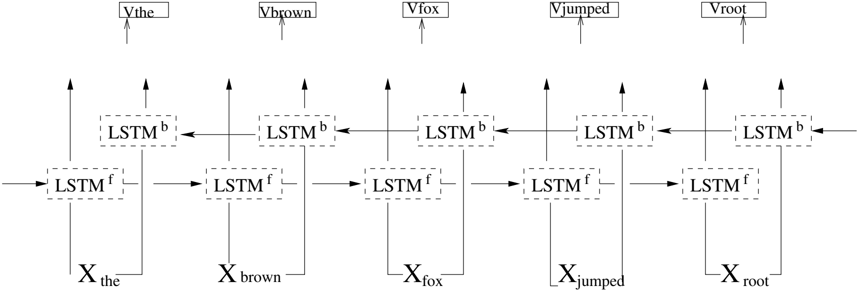
Results: BiLSTM + composition



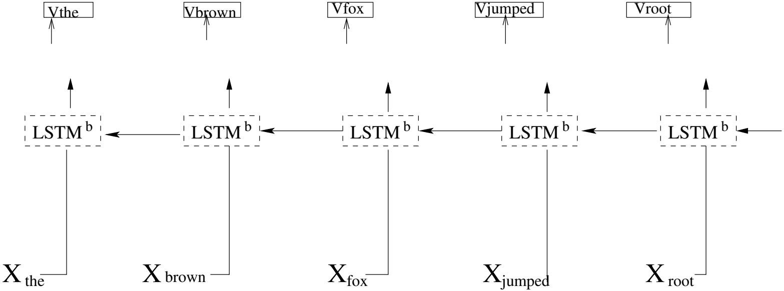
LSTM Feature Extractors



LSTM Feature Extractors

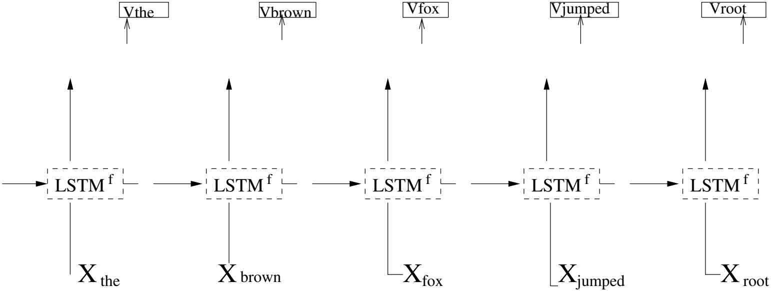


LSTM Feature Extractors



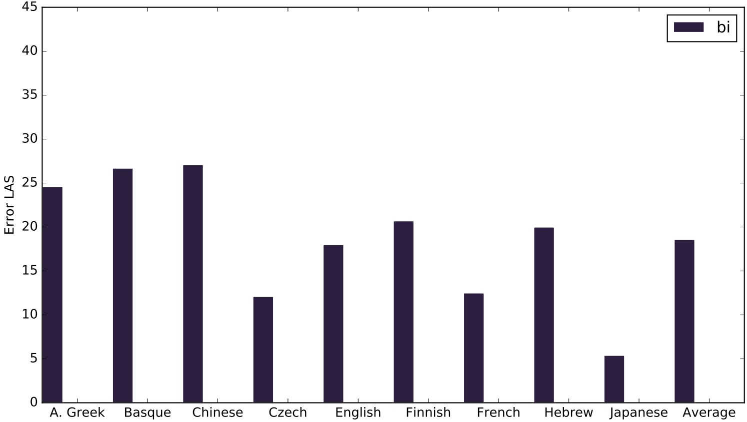
bw

LSTM Feature Extractors

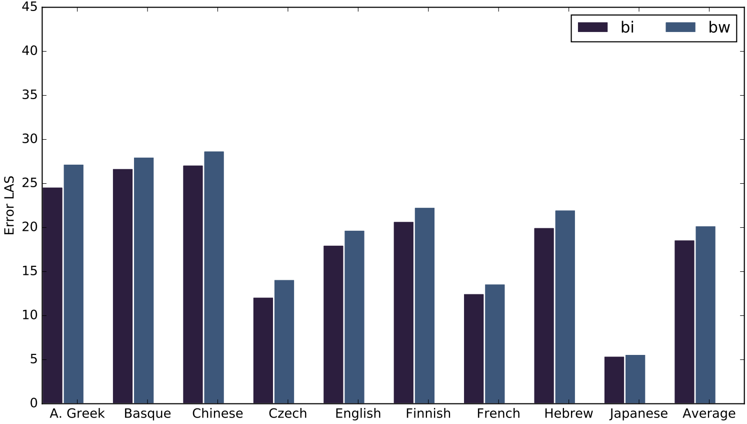


fw

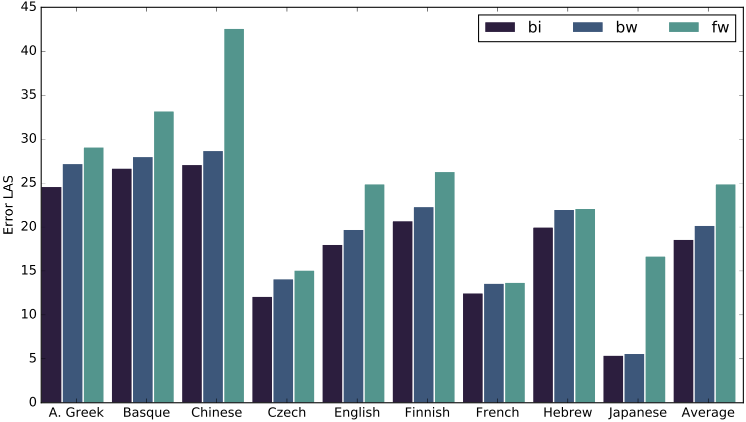
Results: BiLSTM ablations



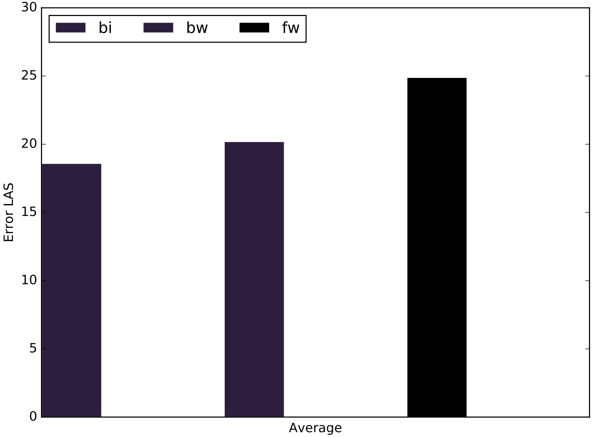
Results: BiLSTM ablations



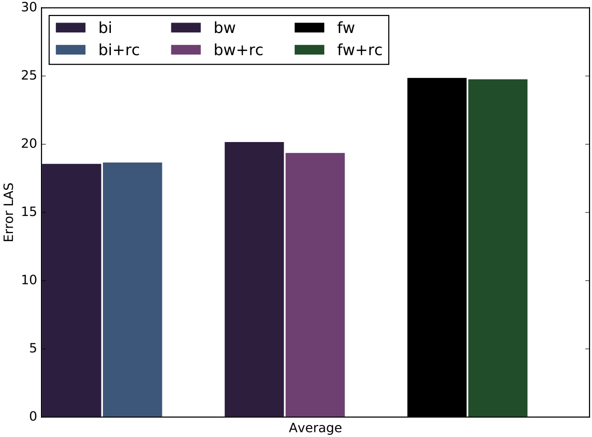
Results: BiLSTM ablations



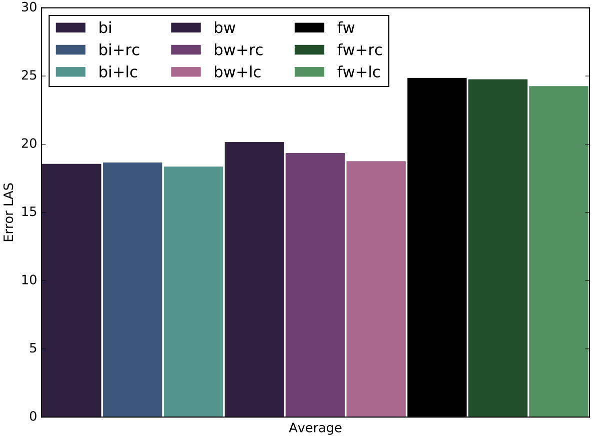
Results: BiLSTM ablations + composition



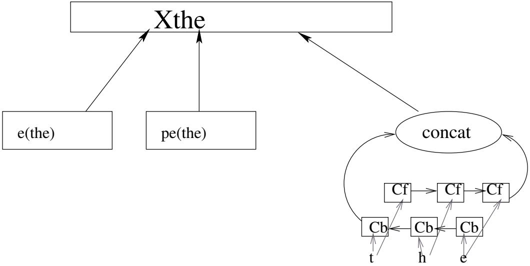
Results: BiLSTM ablations + composition



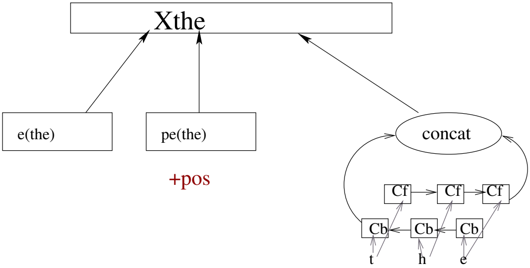
Results: BiLSTM ablations + composition



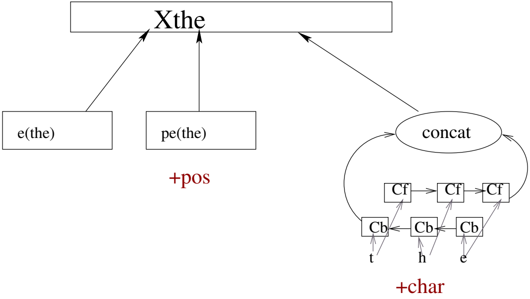
Word representation



Word representation



Word representation



Composition gap recovery

	[bw+lc]-bw	[fw+lc]-fw
pos+char+	1.4	0.6
pos+char-	1.3	0.6
pos-char+	1.6	0.7
pos-char-	2	1

Average

Composition gap recovery

	[bw+lc]-bw	bi-bw	%rec.	[fw+lc]-fw	bi-fw	%rec.
pos+char+	1.4	1.6	87.5	0.6	6.3	9.5
pos+char-	1.3	1.8	72.2	0.6	6.6	9.1
pos-char+	1.6	1.9	84.2	0.7	7.3	9.6
pos-char-	2	3.1	64.5	1	8.7	11.5

Average

Conclusions from this study

- Subtree composition does not reliably help a BiLSTM transition-based parser

Conclusions from this study

- Subtree composition does not reliably help a BiLSTM transition-based parser
- The backward part of the BiLSTM is crucial, especially for right-headed languages

Conclusions from this study

- Subtree composition does not reliably help a BiLSTM transition-based parser
- The backward part of the BiLSTM is crucial, especially for right-headed languages
- The forward part of the BiLSTM is less crucial

Conclusions from this study

- Subtree composition does not reliably help a BiLSTM transition-based parser
- The backward part of the BiLSTM is crucial, especially for right-headed languages
- The forward part of the BiLSTM is less crucial
- A backward LSTM + subtree composition performs close to a BiLSTM

Conclusions from this study

- Subtree composition does not reliably help a BiLSTM transition-based parser
- The backward part of the BiLSTM is crucial, especially for right-headed languages
- The forward part of the BiLSTM is less crucial
- A backward LSTM + subtree composition performs close to a BiLSTM
- POS information and subtree composition are two partially redundant ways of constructing contextual information

Broader perspective

- This study: recursive composition does not help

Broader perspective

- This study: recursive composition does not help
- Falenska and Kuhn (2019): structural features do not help

Broader perspective

- This study: recursive composition does not help
- Falenska and Kuhn (2019): structural features do not help
- Gontrum (2019): attention does not help

Broader perspective

- This study: recursive composition does not help
- Falenska and Kuhn (2019): structural features do not help
- Gontrum (2019): attention does not help

- All information needed is in small set of token representations from stack and buffer

Broader perspective

- This study: recursive composition does not help
- Falenska and Kuhn (2019): structural features do not help
- Gontrum (2019): attention does not help

- All information needed is in small set of token representations from stack and buffer
- Token representations encode subtree information?

Broader perspective

- This study: recursive composition does not help
- Falenska and Kuhn (2019): structural features do not help
- Gontrum (2019): attention does not help

- All information needed is in small set of token representations from stack and buffer
- Token representations encode subtree information?
  - Do we even need parsing algorithms? (Nivre, 2019)

Broader perspective

- This study: recursive composition does not help
- Falenska and Kuhn (2019): structural features do not help
- Gontrum (2019): attention does not help

- All information needed is in small set of token representations from stack and buffer
- Token representations encode subtree information?
  - Do we even need parsing algorithms? (Nivre, 2019)
  - Trees can be decoded directly from BERT contextual embeddings (Hewitt and Manning, 2019)

Broader perspective

Broader perspective

- Linzen et al. (2016) and Gulordava et al. (2018): LM LSTMs learn agreement

Broader perspective

- Linzen et al. (2016) and Gulordava et al. (2018): LM LSTMs learn agreement
- Ravfogel et al. (2018) and Ravfogel et al. (2019): Yes but using local heuristics

Broader perspective

- Linzen et al. (2016) and Gulordava et al. (2018): LM LSTMs learn agreement
- Ravfogel et al. (2018) and Ravfogel et al. (2019): Yes but using local heuristics
- Something like that happening here?

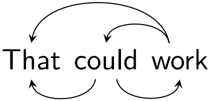
Outline for section 4

Auxiliary Verb Constructions

Dependency Parsing

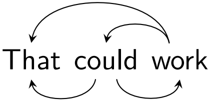
Auxiliary Verb Constructions

Dependency Parsing



Auxiliary Verb Constructions

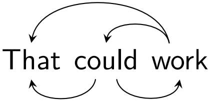
Dependency Parsing



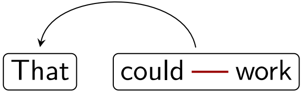
Tesnière

Auxiliary Verb Constructions

Dependency Parsing

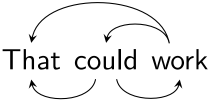


Tesnière

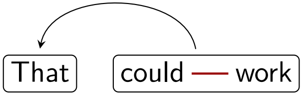


Auxiliary Verb Constructions

Dependency Parsing



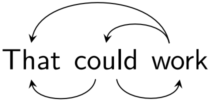
Tesnière



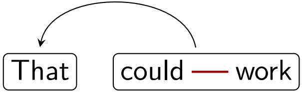
	parsing	Tesnière

Auxiliary Verb Constructions

Dependency Parsing



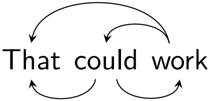
Tesnière



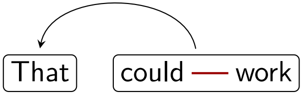
	parsing	Tesnière
Unit of syntax		

Auxiliary Verb Constructions

Dependency Parsing



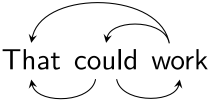
Tesnière



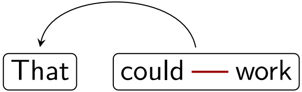
	parsing	Tesnière
Unit of syntax	words	

Auxiliary Verb Constructions

Dependency Parsing



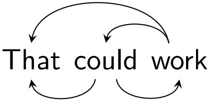
Tesnière



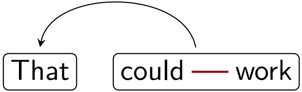
	parsing	Tesnière
Unit of syntax	words	nucleus

Auxiliary Verb Constructions

Dependency Parsing



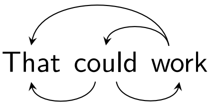
Tesnière



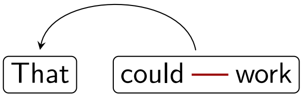
	parsing	Tesnière
Unit of syntax	words	nucleus
Relations between words		

Auxiliary Verb Constructions

Dependency Parsing



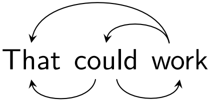
Tesnière



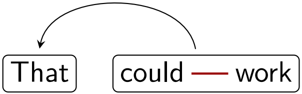
	parsing	Tesnière
Unit of syntax	words	nucleus
Relations between words	dependency	

Auxiliary Verb Constructions

Dependency Parsing



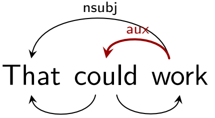
Tesnière



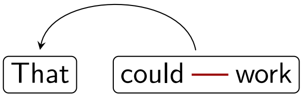
	parsing	Tesnière
Unit of syntax	words	nucleus
Relations between words	dependency	dependency, transfer, junction

Auxiliary Verb Constructions

Dependency Parsing



Tesnière

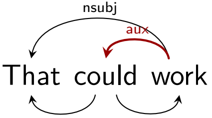


	parsing	Tesnière
Unit of syntax	words	nucleus
Relations between words	dependency	dependency, transfer, junction

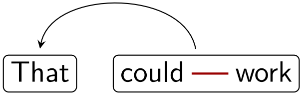
UD compatible with Tesnière

Auxiliary Verb Constructions

Dependency Parsing



Tesnière

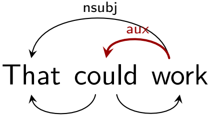


	parsing	Tesnière
Unit of syntax	words	nucleus
Relations between words	dependency	dependency, transfer, junction

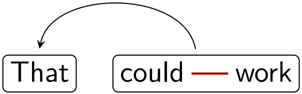
UD compatible with Tesnière
But parsers don't know that

Auxiliary Verb Constructions

Dependency Parsing



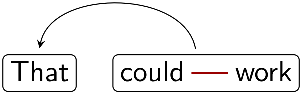
Tesnière



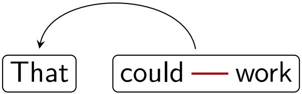
	parsing	Tesnière
Unit of syntax	words	nucleus
Relations between words	dependency	dependency, transfer, junction

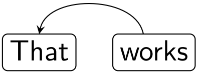
UD compatible with Tesnière
But parsers don't know that
Or do they?

Research questions



Research questions





Research questions

Do LSTM-based parsers learn the notion of dissociated nucleus?

Research questions

Do LSTM-based parsers learn the notion of dissociated nucleus?

Dissociated nucleus $\sim$ nucleus

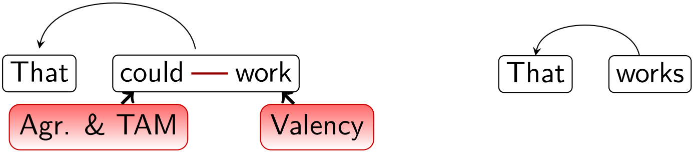
Research questions

Do LSTM-based parsers learn the notion of dissociated nucleus?

Dissociated nucleus $\sim$ nucleus

Diagnostic classifier

Research questions

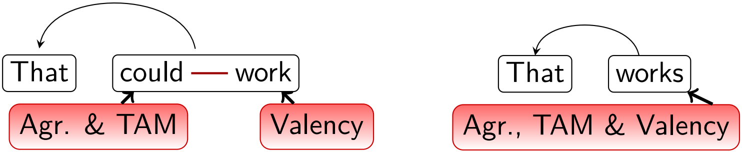


Do LSTM-based parsers learn the notion of dissociated nucleus?

Dissociated nucleus $\sim$ nucleus

Diagnostic classifier

Research questions

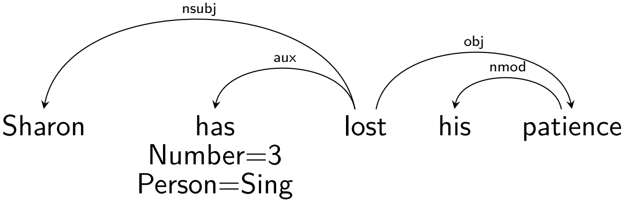


Do LSTM-based parsers learn the notion of dissociated nucleus?

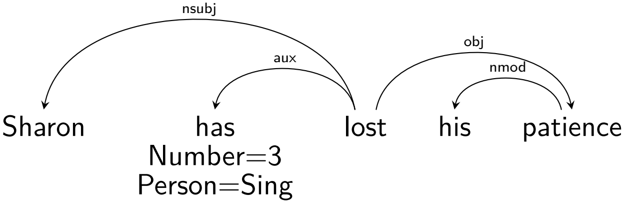
Dissociated nucleus $\sim$ nucleus

Diagnostic classifier

Diagnostic classifier: task

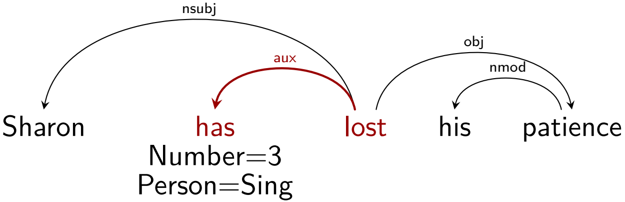


Diagnostic classifier: task



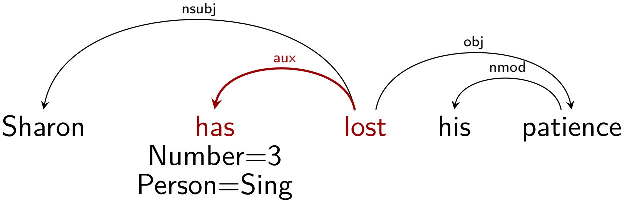
AVC

Diagnostic classifier: task



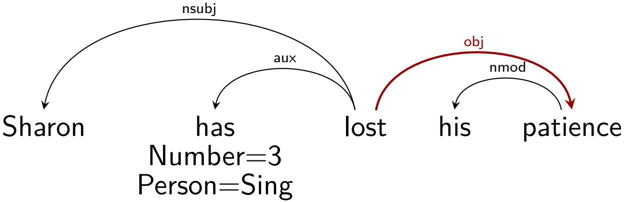
AVC

Diagnostic classifier: task



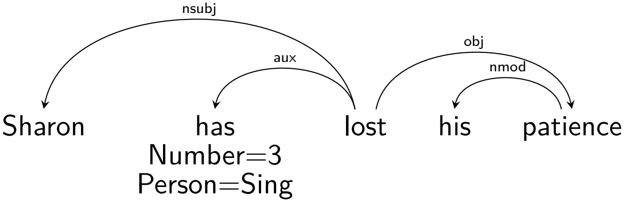
Transitivity: has object? True/False

Diagnostic classifier: task



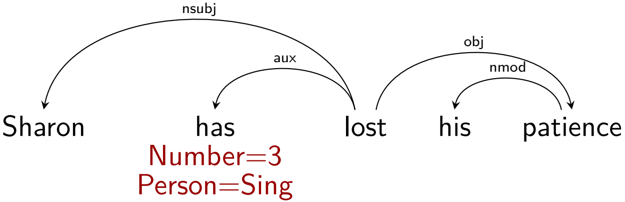
Transitivity: has object? True/False

Diagnostic classifier: task



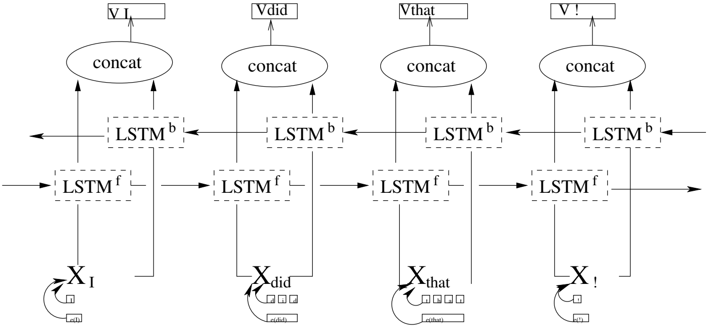
Agreement: Person + Number (sg/pl + 1/2/3)

Diagnostic classifier: task

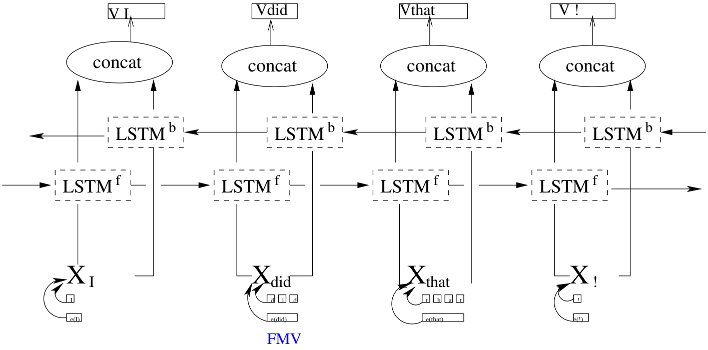


Agreement: Person + Number (sg/pl + 1/2/3)

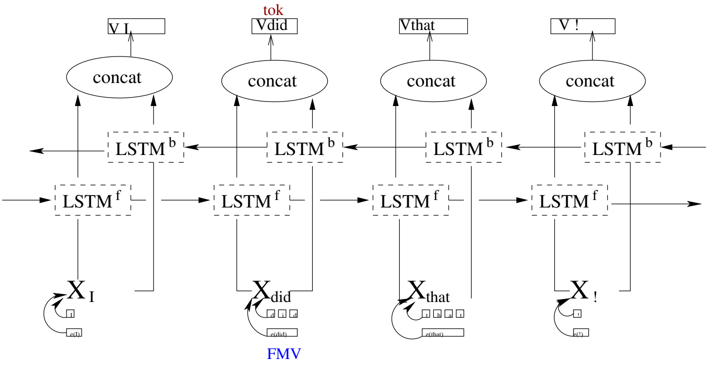
Vectors


V I
Vdid
Vthat
V !
concat
concat
concat
concat
LSTM b
LSTM b
LSTM b
LSTM b
LSTM f
LSTM f
LSTM f
LSTM f
X I
Xdid
Xthat
X !

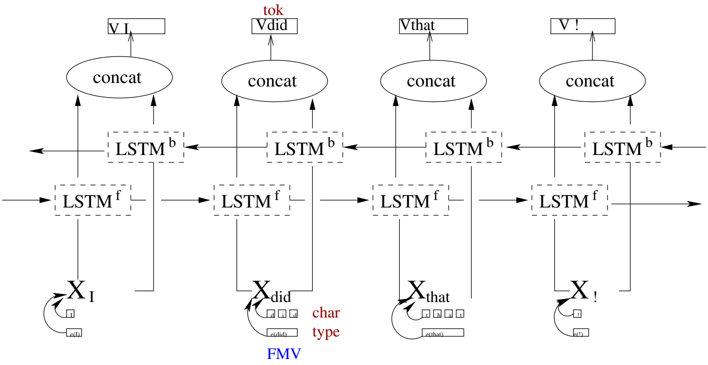
Vectors


V I
Vdid
Vthat
V !
concat
concat
concat
concat
LSTM b
LSTM b
LSTM b
LSTM b
LSTM f
LSTM f
LSTM f
LSTM f
X I
X did
X that
X !
FMV

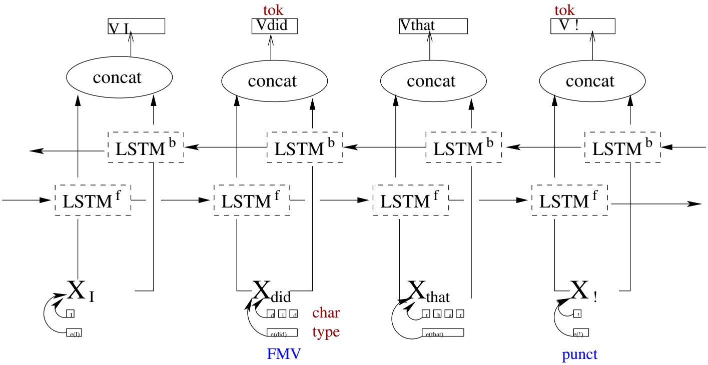
Vectors


tok
V I
Vdid
Vthat
V !
concat
concat
concat
concat
LSTM b
LSTM b
LSTM b
LSTM b
LSTM f
LSTM f
LSTM f
LSTM f
X I
X did
X that
X !
FMV

Vectors


tok
V I
Vdid
Vthat
V !
concat
concat
concat
concat
LSTM b
LSTM b
LSTM b
LSTM b
LSTM f
LSTM f
LSTM f
LSTM f
X I
X did
X that
X !
char
type
FMV

Vectors


tok
tok
V I
Vdid
Vthat
V !
concat
concat
concat
concat
LSTM b
LSTM b
LSTM b
LSTM b
LSTM f
LSTM f
LSTM f
LSTM f
X I
X did
X that
X !
char
type
FMV
punct

Dataset

		FMV		punct		AVC	
		train	dev	train	dev	train	dev
T	Catalan	14K	2K	7K	964	12K	2K
	Croatian	6K	803	4K	491	5K	653
	Dutch	9K	618	6K	516	5K	251
	Finnish	12K	1K	9K	1K	4K	458
A	Catalan	14K	2K	7K	964	12K	2K
	Croatian	6K	803	4K	491	5K	653
	Dutch	9K	618	6K	516	5K	246
	Finnish	10K	1K	8K	850	4K	443

Agreement and Transitivity in FMVs

Agreement

Agreement and Transitivity in FMVs

Agreement

did
FMV

Agreement and Transitivity in FMVs

Agreement

did
FMV

Transitivity

Agreement and Transitivity in FMVs

Agreement

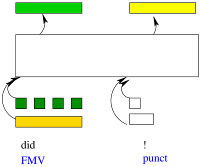
did
FMV

Transitivity

did
FMV

Agreement and Transitivity in FMVs

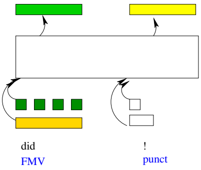
Agreement



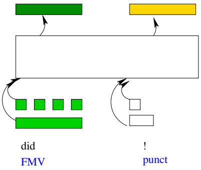
Transitivity

Agreement and Transitivity in FMVs

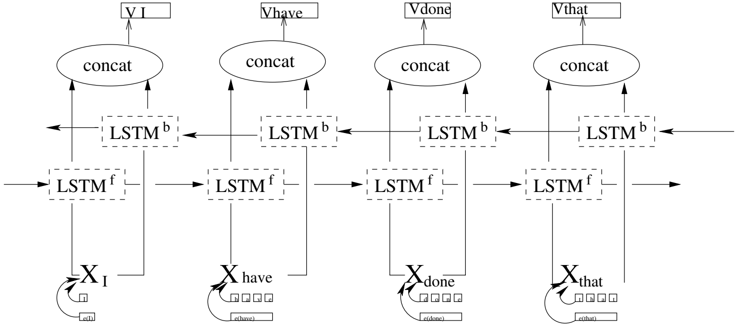
Agreement



Transitivity

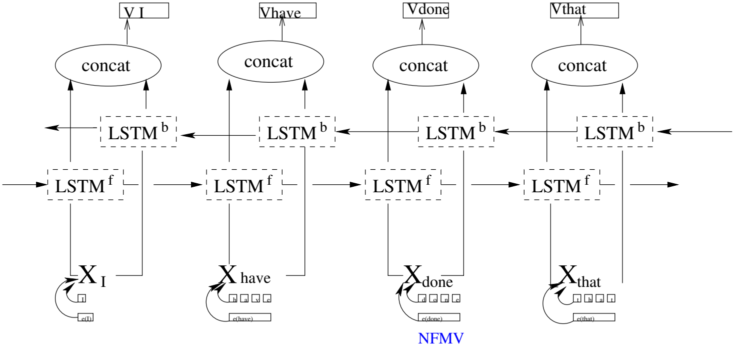


Vectors AVC

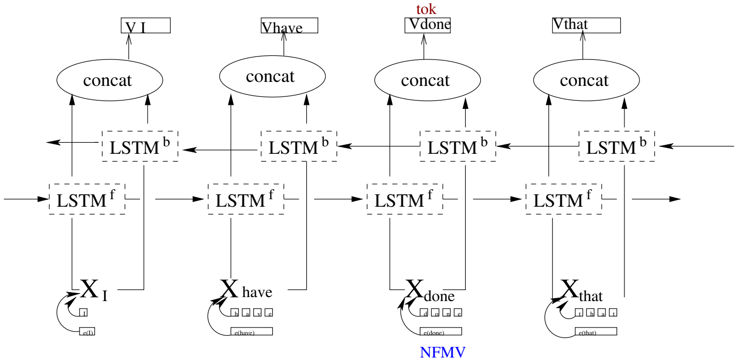


V I
Vhave
Vdone
Vthat
concat
concat
concat
concat
LSTM b
LSTM b
LSTM b
LSTM b
LSTM f
LSTM f
LSTM f
LSTM f
X I
X have
Xdone
Xthat
e(I)
e(have)
e(done)
e(that)

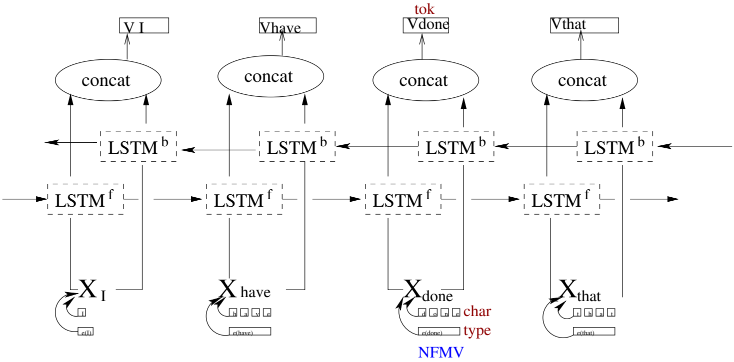
Vectors AVC


V I
Vhave
Vdone
Vthat
concat
concat
concat
concat
LSTM b
LSTM b
LSTM b
LSTM b
LSTM f
LSTM f
LSTM f
LSTM f
X I
X have
X done
X that
NFMV

Vectors AVC

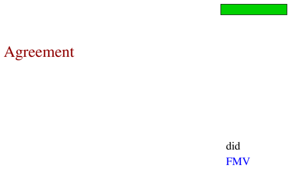

tok
V I
Vhave
Vdone
Vthat
concat
concat
concat
concat
LSTM b
LSTM b
LSTM b
LSTM b
LSTM f
LSTM f
LSTM f
LSTM f
X I
X have
X done
X that
e(I)
e(have)
e(done)
e(that)
NFMV

Vectors AVC



tok
V I
Vhave
Vdone
Vthat
concat
concat
concat
concat
LSTM b
LSTM b
LSTM b
LSTM b
LSTM f
LSTM f
LSTM f
LSTM f
X I
X have
X done
X that
char
type
NFMV

Agreement and Transitivity in AVCs vs FMVs



Agreement and Transitivity in AVCs vs FMVs

Agreement

did
FMV

done
NFMV

Transitivity

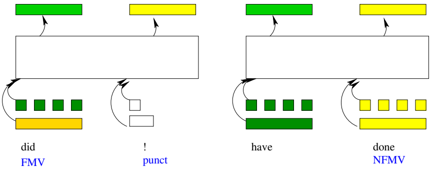
did
FMV

Agreement and Transitivity in AVCs vs FMVs

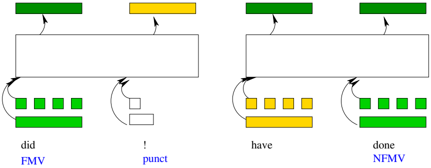
Agreement	did FMV	done NFMV
Transitivity	did FMV	done NFMV

Agreement and Transitivity in AVCs vs FMVs

Agreement

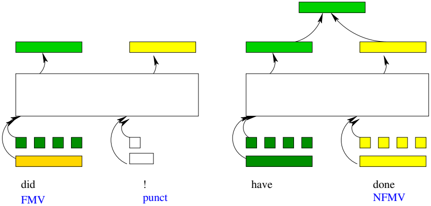


Transitivity

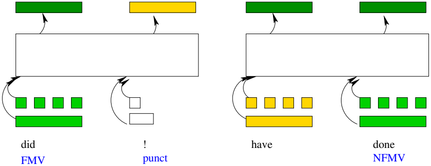


Agreement and Transitivity in AVCs vs FMVs

Agreement

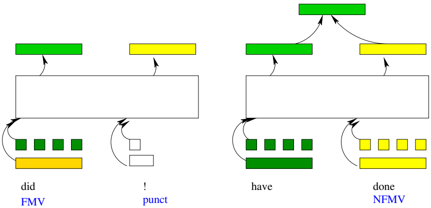


Transitivity

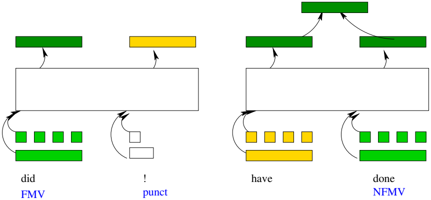


Agreement and Transitivity in AVCs vs FMVs

Agreement



Transitivity



Conclusions from this study

- Our parser does not learn the notion of dissociated nucleus

Conclusions from this study

- Our parser does not learn the notion of dissociated nucleus
- Composition helps learning this

Outline for section 5

General conclusions

Conclusions

- Composition does not help accuracy of a BiLSTM parser

General conclusions

Conclusions

- Composition does not help accuracy of a BiLSTM parser
- Composition might be useful to learn relations of **transfer**

General conclusions

Conclusions

- Composition does not help accuracy of a BiLSTM parser
- Composition might be useful to learn relations of **transfer**

Future Work

- Token vectors encode subtrees or parser uses heuristics?

General conclusions

Conclusions

- Composition does not help accuracy of a BiLSTM parser
- Composition might be useful to learn relations of **transfer**

Future Work

- Token vectors encode subtrees or parser uses heuristics?
- LSTMs vs Transformer

Thanks!

Thanks!

References

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

Agnieszka Falenska and Jonas Kuhn. 2019. The (Non-)Utility of Structural Features in BiLSTM-based Dependency Parsers. In *Proceedings of ACL*.

Johannes Gontrum. 2019. Attention Mechanisms for Transition-based Dependency Parsing. Master's thesis, Uppsala University.

Kristina Gulordava, Piotr Bojanowski, Edouard Grave, Tal Linzen, and Marco Baroni. 2018. Colorless green recurrent networks dream hierarchically. In *Proceedings of NAACL*.

John Hewitt and Christopher D Manning. 2019. A Structural Probe for Finding Syntax in Word Representations. In *Proceedings of NAACL*.

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.

Miryam de Lhoneux, Miguel Ballesteros, and Joakim Nivre. 2019a. Recursive subtree composition in LSTM-based dependency parsing. In *Proceedings of NAACL*.

Miryam de Lhoneux, Yan Shao, Ali Basirat, Eliyahu Kiperwasser, Sara Stymne, Yoav Goldberg, and Joakim Nivre. 2017. From raw text to universal dependencies - look, no tags! In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*.

Miryam de Lhoneux, Sara Stymne, and Joakim Nivre. 2019b. What Should/Do/Can LSTMs Learn When Parsing Auxiliary Verb Constructions? *arXiv preprint arXiv:1907.07950*. Under review.

Tal Linzen, Emmanuel Dupoux, and Yoav Goldberg. 2016. Assessing the ability of LSTMs to learn syntax-sensitive dependencies. *Transactions of the Association for Computational Linguistics*, 4:521–535.

Joakim Nivre. 2019. Is the End of Supervised Parsing in Sight? Twelve Years Later. Invited talk at EurNLP.

Shauli Ravfogel, Yoav Goldberg, and Tal Linzen. 2019. Studying the inductive biases of RNNs with synthetic variations of natural languages. In *Proceedings of NAACL*.

Shauli Ravfogel, Yoav Goldberg, and Francis Tyers. 2018. Can LSTM Learn to Capture Agreement? The Case of Basque. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.