Evaluating LSTM Models for Grammatical Function Labelling

Bich Ngoc Do♢ and Ines Rehbein♣

Leibniz ScienceCampus
Universität Heidelberg♢
Institut für Deutsche Sprache Mannheim♣

September 22, 2017

Grammatical Function Labelling

- Grammatical function (GF) labels help to interpret a sentence's meaning.

Grammatical Function Labelling

- Grammatical function (GF) labels help to interpret a sentence's meaning.
- Challenge: *case syncretism*. E.g: German

Grammatical Function Labelling

- Grammatical function (GF) labels help to interpret a sentence's meaning.
- Challenge: *case syncretism*. E.g: German

Die Frau$_{Nom/Acc}$	beißt	das Pferd$_{Nom/Acc}$	.
The woman	bites	the horse	.

Grammatical Function Labelling

- Grammatical function (GF) labels help to interpret a sentence's meaning.
- Challenge: *case syncretism*. E.g: German

  Die Frau$_{Nom/Acc}$ beißt das Pferd$_{Nom/Acc}$.

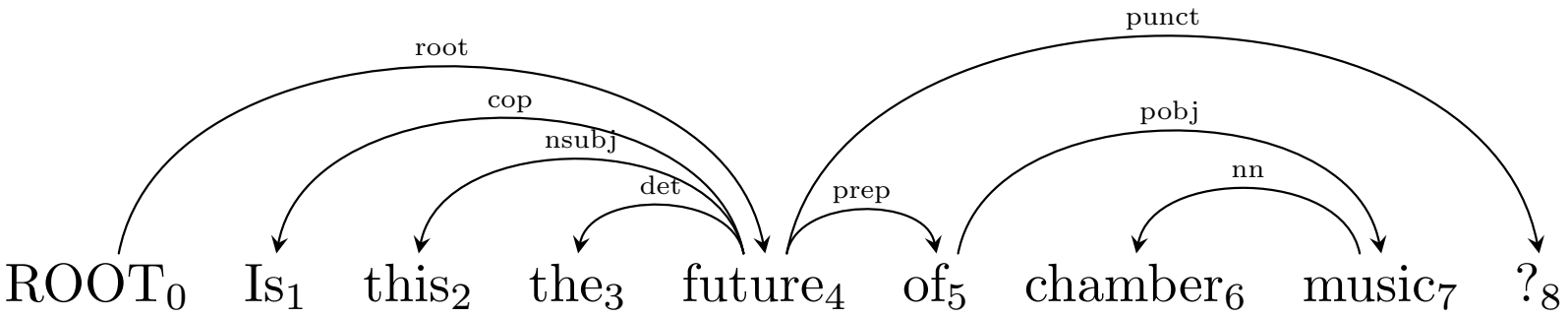
  The woman bites the horse .

  "The women bites the horse. / The horse bites the women."

Related Work

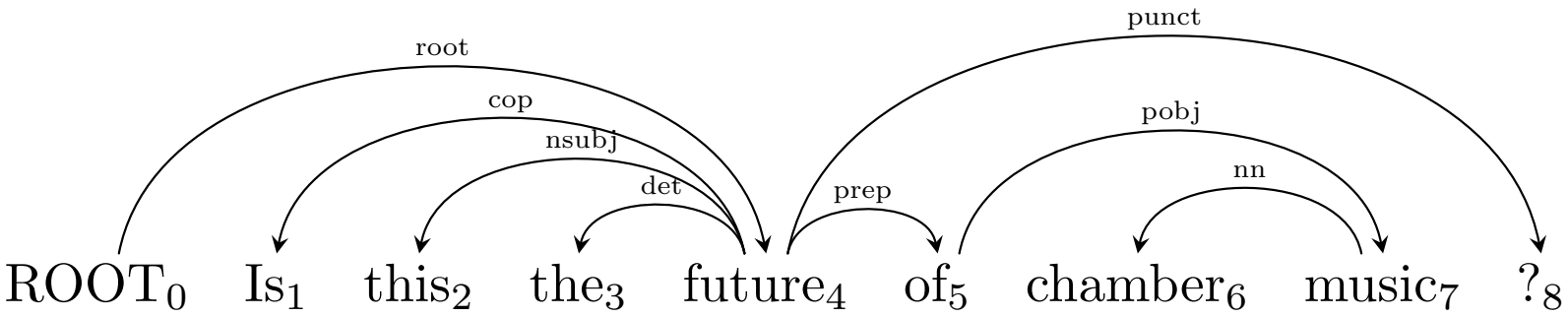
- Most studies assign GF labels to constituency trees (Klenner, 2007; Chrupała and van Genabith, 2006; Seeker et al., 2010)
- Only a few studies model GF labelling as a separate task in dependency parsing:
  - McDonald et al. (2006): label all children of a node in a sequence labelling task using CRFs
  - Zhang et al. (2017): use a two-layer rectifier network to assign a label to each head-dependent pair

Labelling Dependencies with History



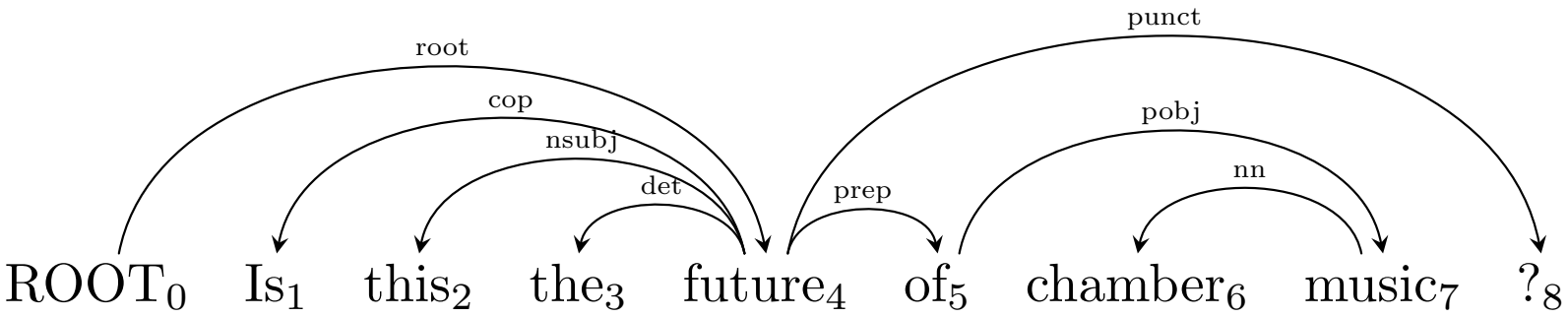
- Labelling benefits from context: the parent and grandparent nodes or the siblings...

Labelling Dependencies with History



- Labelling benefits from context: the parent and grandparent nodes or the siblings...
- Well known errors from local parsing models: duplicate subjects

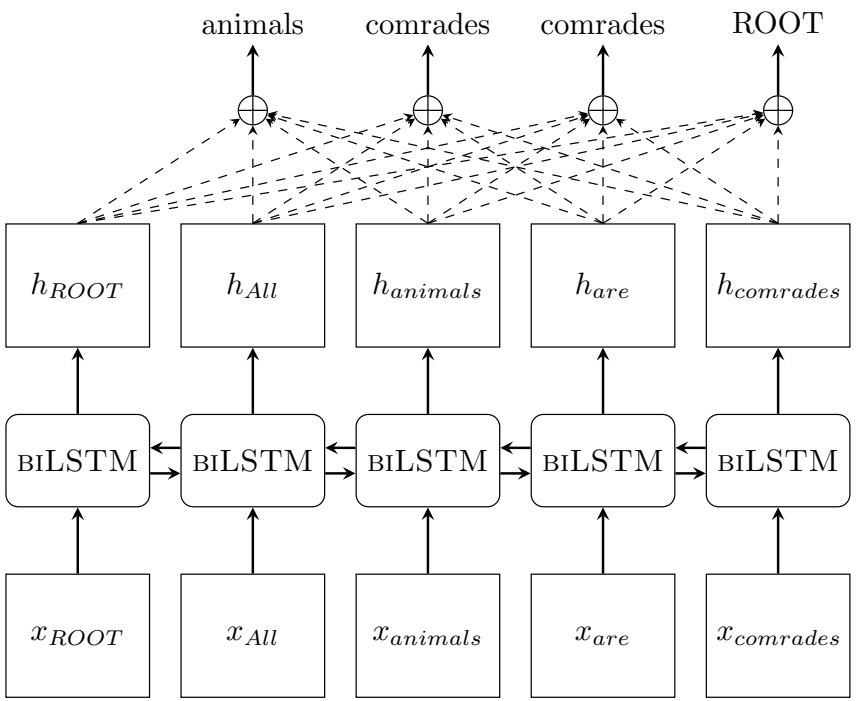
Labelling Dependencies with History



- Labelling benefits from context: the parent and grandparent nodes or the siblings...
- Well known errors from local parsing models: duplicate subjects

$\Rightarrow$ Augment the labeller with different LSTM architectures.

Experimental Framework: DeNSe (Zhang et al., 2017)



- Uses a bidirectional LSTM to encode each word in a sentence
- Parsing in two steps
- Input for labelling the edge between head w_i and child w_j is $[b_i; b_j]$ where:

$$b_i = [x_i; h_i^F, h_i^B]$$

Label Prediction as a Sequence Labelling Task

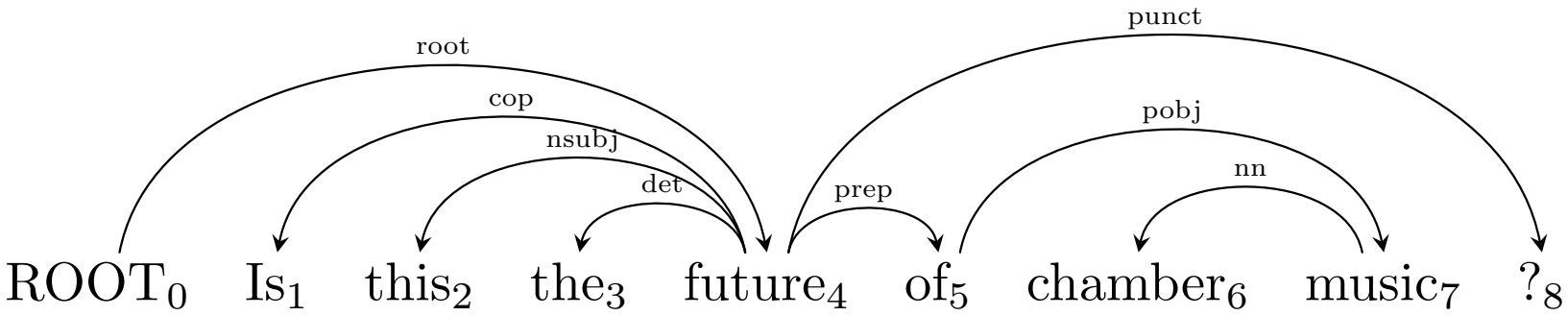
- McDonald et al. (2006) considered all children of a node.

Label Prediction as a Sequence Labelling Task

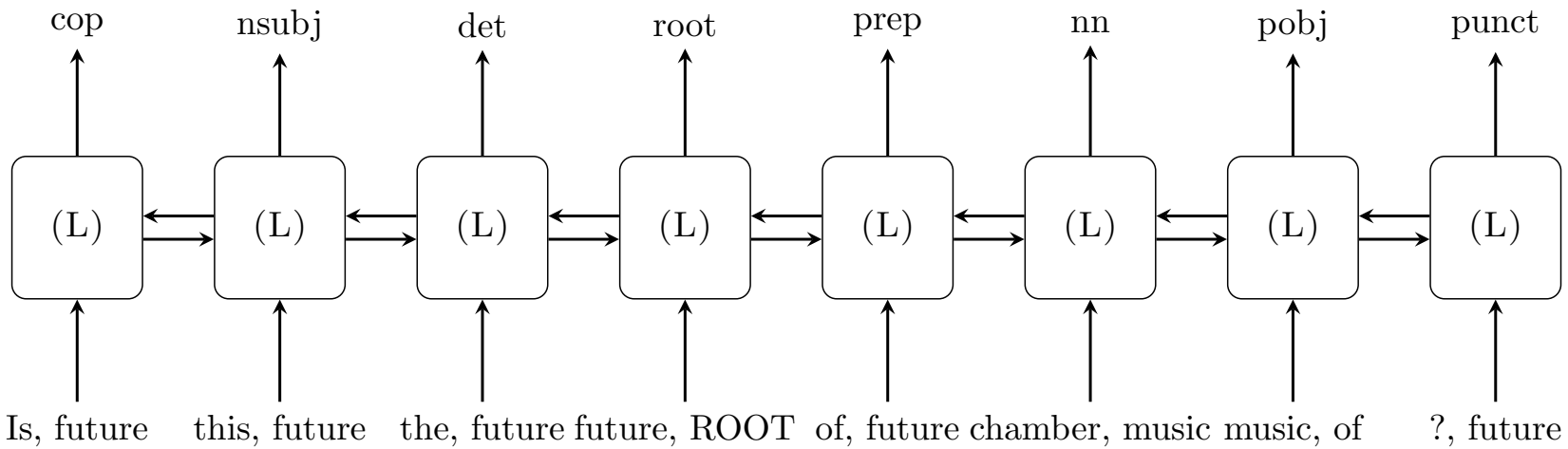
- McDonald et al. (2006) considered all children of a node.
- We consider all label decisions and feed them to a bidirectional LSTM: given a sequence of words $S = (w_1, ..., w_N)$ and their corresponding head $(h_1, ..., h_N)$:

$$h_i^{F(\text{lbl})} = \text{LSTM}_{\text{lbl}}^{F}(b_i, b_{h_i}, h_{i-1}^{F(\text{lbl})})$$
$$h_i^{B(\text{lbl})} = \text{LSTM}_{\text{lbl}}^{B}(b_i, b_{h_i}, h_{i+1}^{B(\text{lbl})})$$

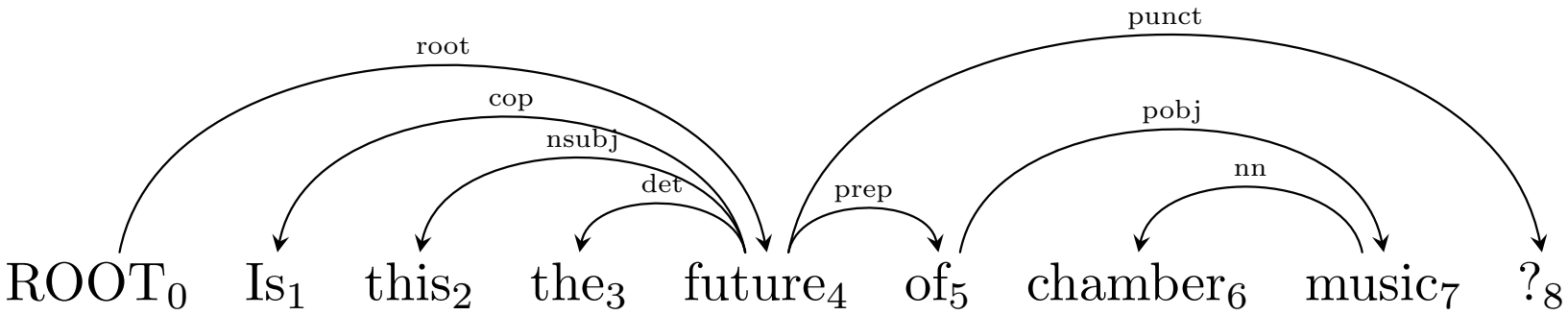
Linear LSTMs



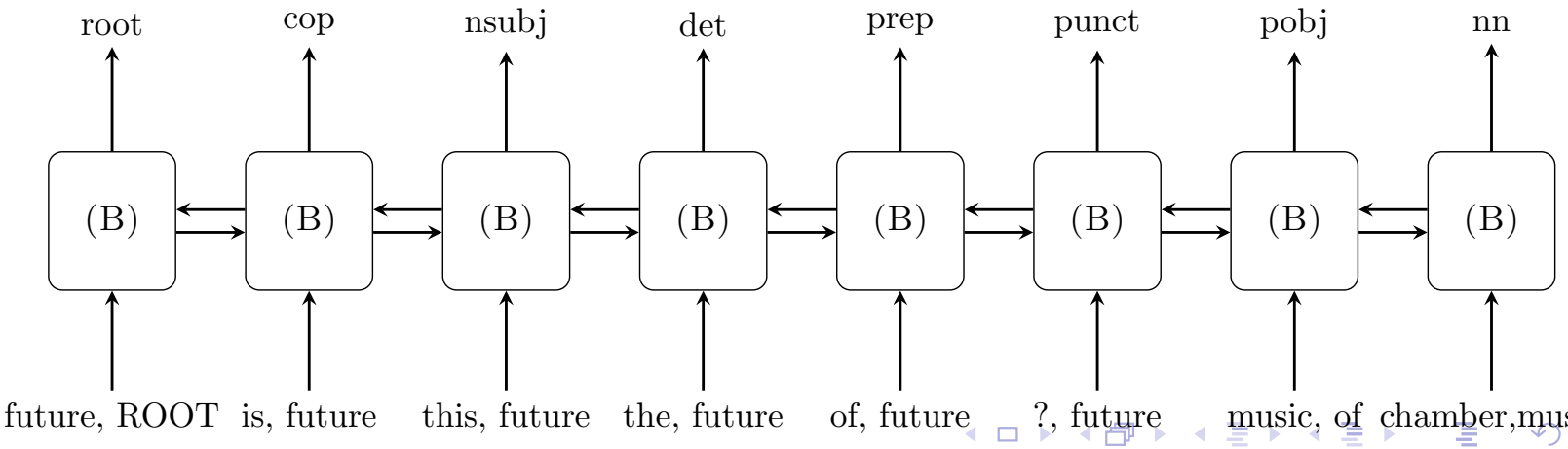
- BILSTM(L): Tree nodes are ordered according to their surface order in the sentence (linear order).



Linear LSTMs



- BILSTM(B): Tree nodes are ordered according to a breadth-first traversal (BFS) of the tree, starting from the root node.

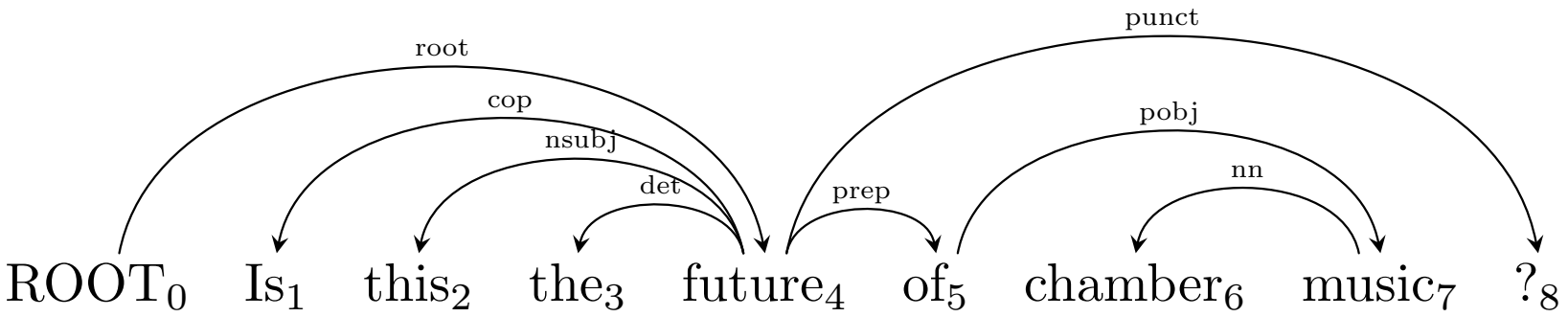


Top-down Tree LSTMs

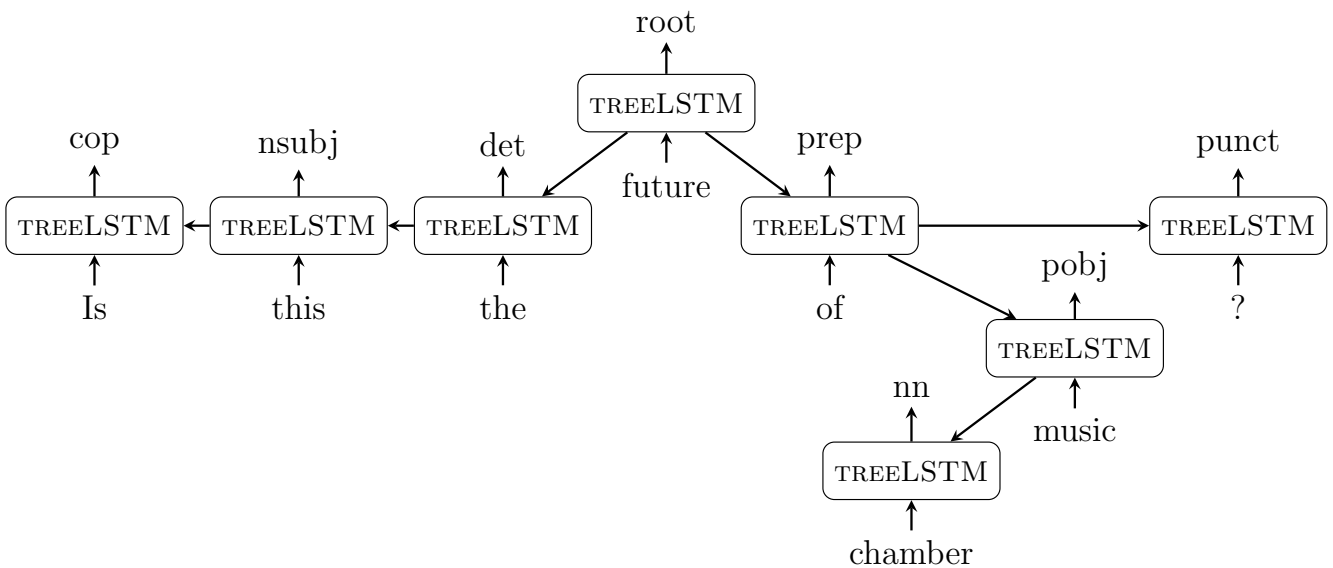
- Top-down tree LSTMs (Zhang et al., 2016):
  - Use 1 (instead of 4) LSTM
  - Do not stack LSTMs
- Hidden state:

$$h_i^{(lbl)} = \text{treeLSTM}(b_i, h_{i-1}^{(lbl)})$$

Top-down Tree LSTMs



- TREELSTM:



Top-down Tree LSTMs

- Notes:
  - The input to the LSTM is the hidden representation of a node, not a pair
  - The tree model also has a *shorter history chain* and information only flows in *one direction*

Data

Language	Morphology	Word order	Dataset	Test size
German				

Data

Language	Morphology	Word order	Dataset	Test size
German	Rich(er)	Semi-free	CoNLL 2006	357 sent.

Data

Language	Morphology	Word order	Dataset	Test size
English	Poor	Configurational	PTB	2416 sent.
German	Rich(er)	Semi-free	CoNLL 2006	357 sent.

Data

Language	Morphology	Word order	Dataset	Test size
English	Poor	Configurational	PTB	2416 sent.
German	Rich(er)	Semi-free	CoNLL 2006	357 sent.
Czech	Rich	Free	CoNLL 2006	365 sent.

Data

Language	Morphology	Word order	Dataset	Test size
English	Poor	Configurational	PTB	2416 sent.
German	Rich(er)	Semi-free	CoNLL 2006	357 sent.
			SPMRL 2014	5,000 sent.
Czech	Rich	Free	CoNLL 2006	365 sent.

Setup

- First train the unlabelled parsing models, then train different labellers (2 linear LSTMs, 1 tree LSTM) while fixing the unlabelled parameters
- Do not use any pre-trained embeddings
- Report unlabelled attachment score (UAS) and labelled attachment score (LAS) (excluding punctuations)

Results of Different Labellers

Model	**en**	**cs**	**de**$_{CoNLL}$	**de**$_{SPMRL}$
UAS	93.35	89.70	93.09	91.29
Baseline	91.58	83.42	90.22	88.15
BILSTM(L)	**91.92***	**84.08***	90.87*	88.73*
BILSTM(B)	91.91*	83.80	**90.97***	**88.74***
TREELSTM	**91.92***	83.82	90.89*	**88.74***
DENSE	91.90	81.72	89.60	-

Table: (*) indicates that the difference between the model and the baseline is statistically significant ($p < .001$)

Compare to the SPMRL 2014 Winning Systems

- Our best results are only 0.3% lower than the wining system (Björkelund et al., 2014) *without reranker (blended)*.

Compare to the SPMRL 2014 Winning Systems

- Our best results are only 0.3% lower than the wining system (Björkelund et al., 2014) *without reranker (blended)*.
- When applied on the output of the *blended* system, LAS slightly improves from 88.62% to 88.76% (TREELSTM).

Compare to the SPMRL 2014 Winning Systems

- Our best results are only 0.3% lower than the wining system (Björkelund et al., 2014) *without reranker (blended)*.
- When applied on the output of the *blended* system, LAS slightly improves from 88.62% to 88.76% (TREELSTM).
- When applied on *unlabelled gold trees*, the distance between our best history-based model and the baseline increases by 1%.

Impact on Core GFs

de$_{SPMRL}$	**SB**	**OA**	**DA**	**PD**
	# 6,638	# 3,184	# 568	# 1,045
baseline	90.3	83.6	64.7	77.1
BILSTM(L)	91.4	85.3	67.7	80.0
BILSTM(B)	**91.9**	**85.4**	**69.3**	**80.5**
treeLSTM	91.2	85.1	68.6	79.8
de$_{SPMRL}$	**AG**	**PG**	**OC**	**OG**
	# 2,241	# 388	# 3,652	# 21
baseline	91.3	80.0	90.1	0
BILSTM(L)	91.3	81.6	90.5	16.0
BILSTM(B)	**91.5**	**82.4**	**90.7**	**37.0**
treeLSTM	91.4	81.4	90.2	27.6

Table: SB: subj, OA: acc.obj, DA: dat.obj, PD: pred, AG: gen.attribute, PG: phrasal genitive, OC: clausal obj, OG: gen.obj.

Long Dependencies vs. Head Direction

History-based models is *not* better at handling of *long dependencies*, but in dealing with the *uncertainty* in *head direction*.

Conclusions

- Our proposed models are practically simple and computationally inexpensive (as compared to global training or inference), but still do significantly improve labelling performance.

Conclusions

- Our proposed models are practically simple and computationally inexpensive (as compared to global training or inference), but still do significantly improve labelling performance.
- History is especially important for languages with more word order variation.

Conclusions

- Our proposed models are practically simple and computationally inexpensive (as compared to global training or inference), but still do significantly improve labelling performance.
- History is especially important for languages with more word order variation.
- Presenting the input in a BFS order outperforms other LSTM models on core grammatical functions.

Thank you!

Long Dependencies vs. Head Direction

	GF	**en**	**cs**	**de**$_{SPMRL}$
dep-length	sb	3.1	3.4	3.9
	dobj	2.5	*2.4	4.2
	iobj	1.7	-	4.7
left-head ratio	sb	4.6	32.5	34.2
	dobj	97.4	*77.5	37.2
	iobj	100.0	-	27.5

Table: Avg. dependency length and ratio of left arcs vs. all (left + right) arc dependencies for args. (*) in the Czech data, *Obj* subsumes all types of objects, not only direct objects

Björkelund, Anders et al. (2014). "Introducing the IMS-Wroclaw-Szeged-CIS entry at the SPMRL 2014 Shared Task: Reranking and Morpho-syntax meet Unlabeled Data". In: *Proceedings of the First Joint Workshop on Statistical Parsing of Morphologically Rich Languages and Syntactic Analysis of Non-Canonical Languages*. Dublin, Ireland: Dublin City University, pp. 97–102. URL: http://www.aclweb.org/anthology/W14-6110.

Chrupała, Grzegorz and Josef van Genabith (2006). "Using Machine-learning to Assign Function Labels to Parser Output for Spanish". In: *Proceedings of the COLING/ACL on Main Conference Poster Sessions*. COLING-ACL '06. Sydney, Australia, pp. 136–143.

Klenner, Manfred (2007). "Shallow Dependency Labeling". In: *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*. ACL '07. Prague, Czech Republic, pp. 201–204.

McDonald, Ryan, Kevin Lerman, and Fernando Pereira (2006). "Multilingual Dependency Analysis with a Two-stage Discriminative Parser". In: *Proceedings of the 10th Conference on Computational Natural Language Learning*. CoNLL-X '06. New York City, New York, pp. 206–210.

Seeker, Wolfgang et al. (2010). "Hard Constraints for Grammatical Function Labelling". In: *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. ACL '10. Uppsala, Sweden, pp. 1087–1097.

Zhang, Xingxing, Liang Lu, and Mirella Lapata (2016). "Top-down Tree Long Short-Term Memory Networks". In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 310–320. URL: http://www.aclweb.org/anthology/N16-1035.

Zhang, Xingxing, Jianpeng Cheng, and Mirella Lapata (2017). "Dependency Parsing as Head Selection". In: *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. EACL'17. Valencia, Spain, pp. 665–676.