Modeling Input Uncertainty in A Neural Network Dependency Parser.

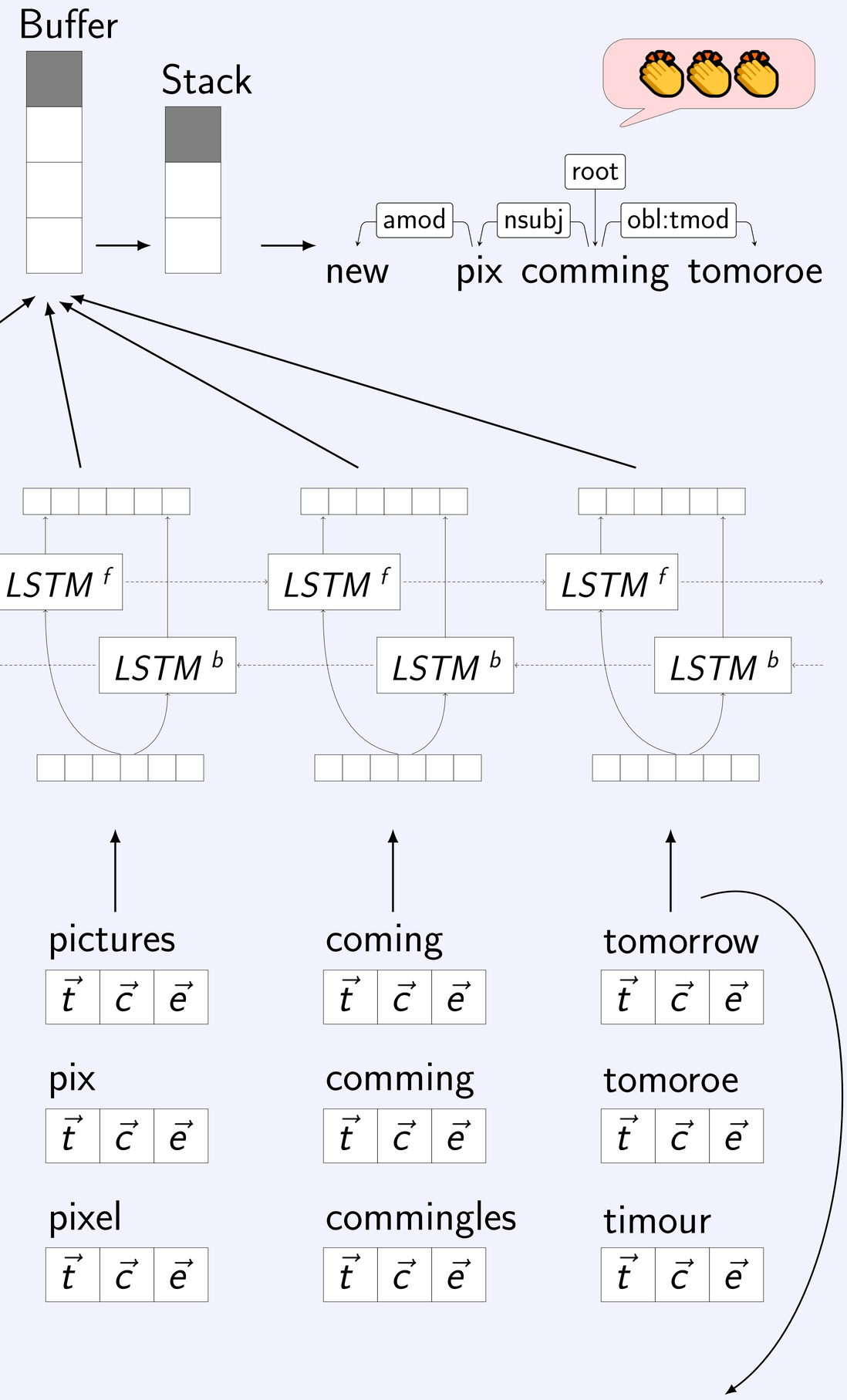
rijksuniversiteit groningen

Rob van der Goot & Gertjan van Noord

NUANCE FOUNDATION

Questions

- Is using normalization beneficial for neural network dependency parsers? (1-BEST)
- Can we improve parser performance by exploiting the top-N normalization candidates? (N-BEST)
- What is the theoretical upperbound of using normalization? (GOLD)



New Treebank

- All tweets from Owoputi and LexNorm which are still available (10,005 tokens)
- Annotated tokenization, Normalization, POS tags and dependency structure
- Also version available with predicted normalization
- Guidelines similar to PoSTWITA-UD (Sanguinetti et al., 2018), UD-TwitterAAE (Blodgett et al., 2018) and Tweebank v2 (Liu et al., 2018)

Settings

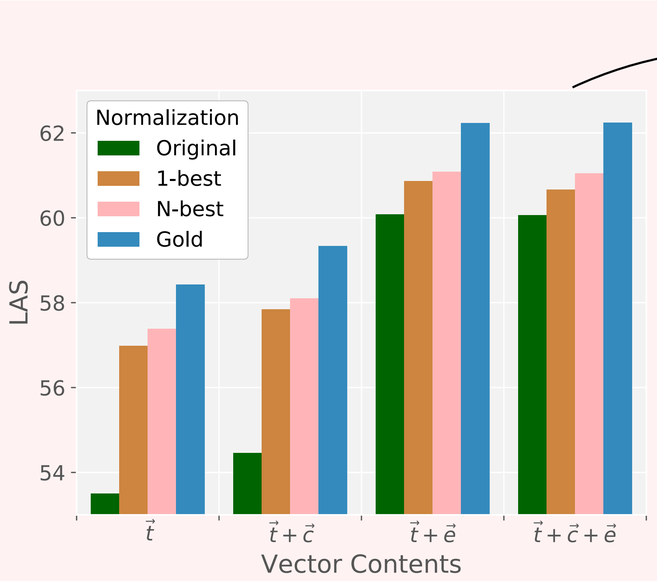
ORIGINAL: $\vec{v}_i = \vec{w}_i$

1-BEST: $\vec{v}_i = \vec{n}_{i0}$

N-BEST: $\vec{v}_i = \sum_{j=0}^{n} p_{ij} * \vec{n}_{ij}$

GOLD: $\vec{v}_i = \vec{g}_i$

Results



Model	UAS	LAS
ORIGINAL	69.63	59.64
1-BEST	70.51	61.76*
N-BEST	70.62	62.30*
GOLD	70.71	62.33

Conclusions

- Using normalization directly is useful for dependency parsing, even when exploiting external and character embeddings
- Integrating normalization results in even higher performance
- When using normalization and external embeddings, character embeddings do not improve results

Source code

www.bitbucket.org/robvanderg/normpar