Modeling Input Uncertainty in A Neural Network Dependency Parser.

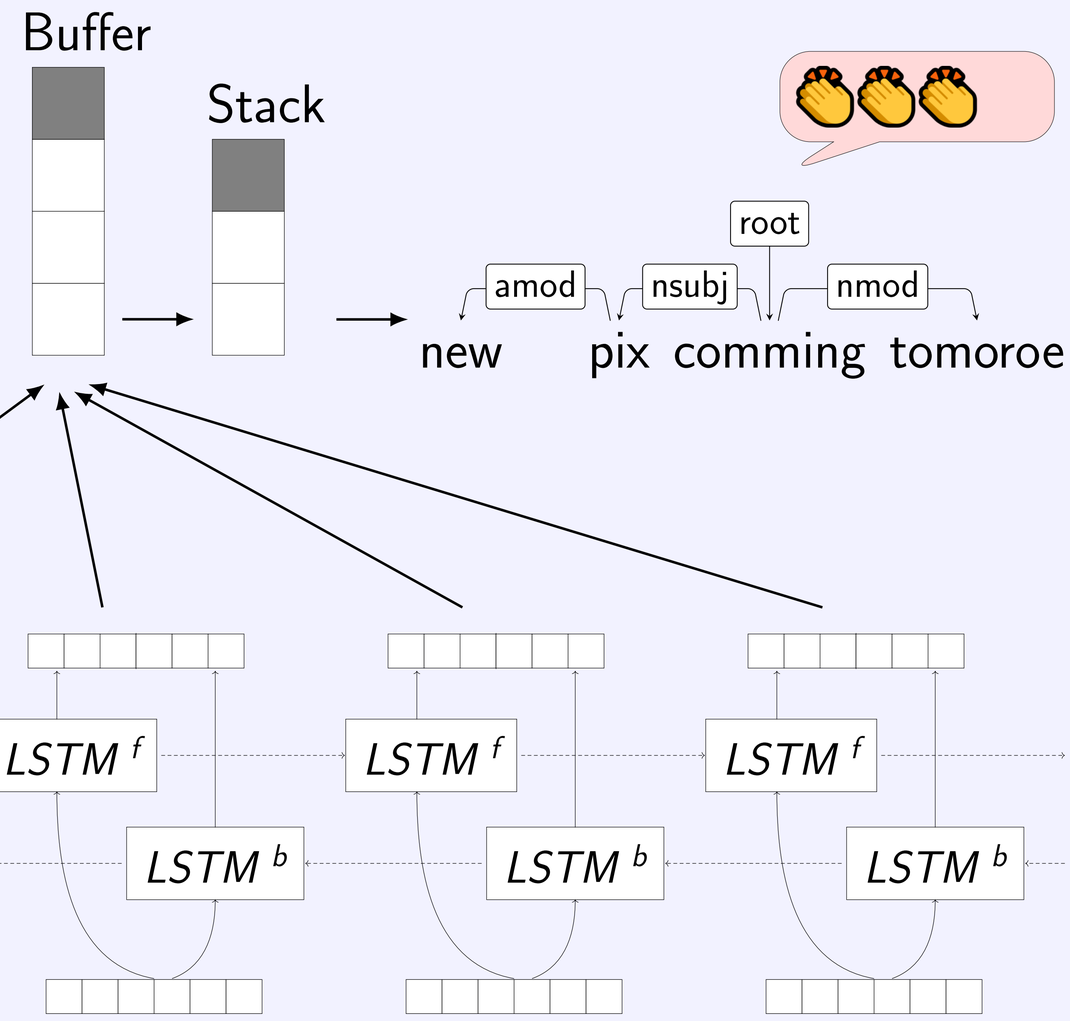
rijksuniversiteit groningen

Rob van der Goot & Gertjan van Noord

NUANCE FOUNDATION

Questions

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



Treebank

- All tweets from Owoputi and LexNorm which are still available
- 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

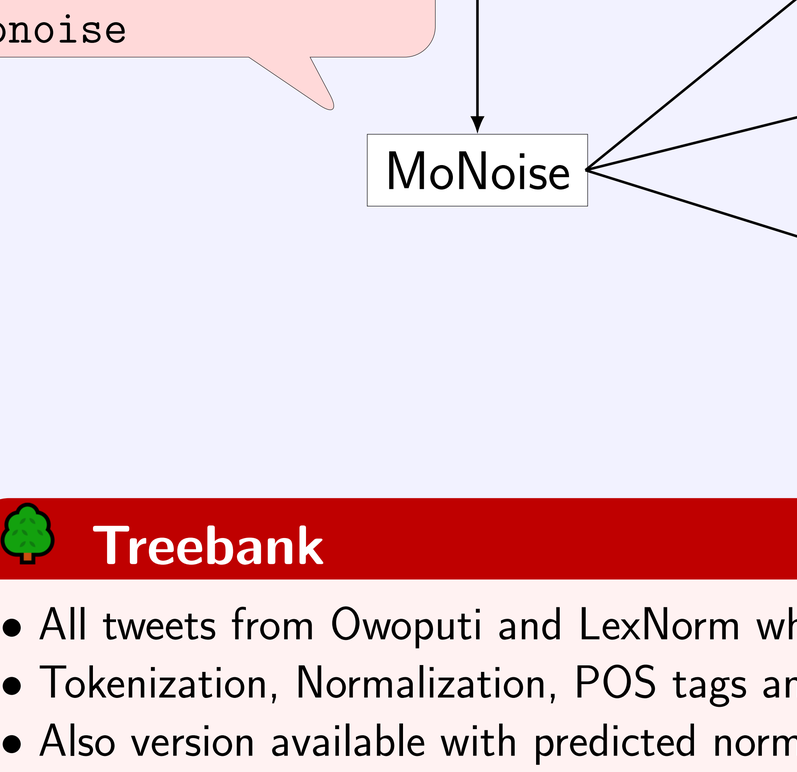
ORIG: $\vec{v}_i = \vec{w}_i$

NORM: $\vec{v}_i = \vec{n}_{i0}$

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

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

Results



Model	UAS	LAS
ORIG	69.63	59.64
NORM	70.51	61.76*
INTEGRATED	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
* New Twitter treebank
* Source code: www.bitbucket.org/robvanderg/normpar