

*Multitask Pointer Network for Discontinuous Constituent  
Parsing*

*An application to the German language*

James Yu  
jby21@cam.ac.uk

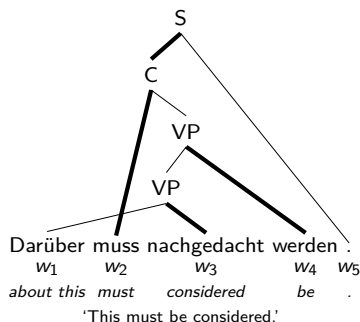
Department of Engineering  
University of Cambridge

16th November 2024

## *Why use a neural network to model German grammar?*

- Constituent trees are a syntactic formalism representing phrasal hierarchy in a sentence.
- Free-order languages like German contain many grammatical discontinuities.
- Discontinuous representations introduce computational complexity but can be more valuable for downstream applications.
- Grammar-less neural network-based models have continually pushed the state of the art.
- My model is based on Fernández-González and Gómez-Rodríguez (2022), who propose an architecture based on pointer neural networks in a multi-task setting.
- I achieve state-of-the-art performance across several metrics on the TIGER treebank.

## Linguistic preliminaries: constituent trees



Let  $w = (w_1, \dots, w_L)$  be a sentence.

### Definition

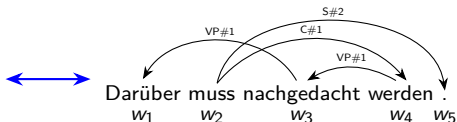
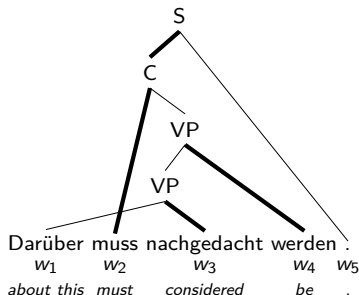
- A *constituent tree* is a rooted tree whose leaves are the words  $(w_i)_{i=1}^L$  and internal nodes are constituents satisfying some constraints.
- A *constituent* is a triple  $(Z, \mathcal{Y}, h)$  containing, respectively, its label, yield, and lexical head.
- A constituent is *discontinuous* if its yield is not contiguous.

## Reduction to dependency parsing

### Definition

A *dependency tree* is a rooted tree spanning the words in the sentence  $(w_i)_{i=1}^L$ . Each edge is labelled and connects a *head word* (parent) to a *dependency* (child).

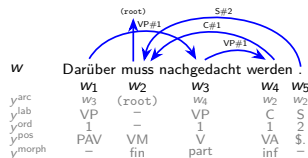
Fernández-González and Martins (2015) show that constituent trees are isomorphic to dependency trees in which the edges contain information about constituent labels and attachment order.



*Figure:* Dependency tree

# Mathematical formalisation

- Regressor: sentence  $(w_i)_{i=1}^L$ .
- Regressand:  $y_i = (y_i^{\text{arc}}, y_i^{\text{lab}}, y_i^{\text{ord}}, y_i^{\text{pos}}, y_i^{\text{morph}})$  for each  $i = 1, \dots, L$ .
- Bottom-up approach: think of arcs going from every child  $w_i$  to its parent  $y_i^{\text{arc}} \in w \setminus w_i$ .



Denote  $y = (y_i)_{i=1}^L$  and with mild abuse of notation let  $y_{<i} = (y_1, \dots, y_{i-1})$  for each  $i = 2, \dots, L$  and  $y_{<1} = 0$ .

## Assumption (conditional independence)

For each  $i = 1, \dots, L$ , the random variables  $y_i^{\text{lab}}, y_i^{\text{ord}}, y_i^{\text{pos}}$  and  $y_i^{\text{morph}}$  are mutually independent conditional on  $y_i^{\text{arc}}, y_{<i}$  and  $w$ .

We can decompose the conditional probability of  $y$  given  $w$ :<sup>1</sup>

$$\begin{aligned}
 p_w(y) &= \prod_{i=1}^L p_w(y_i \mid y_{<i}) \\
 &= \prod_{i=1}^L \left\{ p_w(y_i^{\text{arc}} \mid y_{<i}) p_w(y_i^{\text{lab}} \mid y_i^{\text{arc}}, y_{<i}) \right. \\
 &\quad \cdot p_w(y_i^{\text{ord}} \mid y_i^{\text{arc}}, y_{<i}) p_w(y_i^{\text{pos}} \mid y_i^{\text{arc}}, y_{<i}) p_w(y_i^{\text{morph}} \mid y_i^{\text{arc}}, y_{<i}) \left. \right\}.
 \end{aligned}$$

<sup>1</sup>For notational simplicity we have written  $p_w(\cdot) \equiv p(\cdot \mid w)$  and  $p_w(\cdot \mid \cdot) \equiv p(\cdot \mid \cdot, w)$ .

## Sequence-to-sequence setup

Given an input sentence  $w = (w_i)_{i=1}^L$  we generate *embeddings*  $\omega = (\omega_i)_{i=1}^L$ , where

$$\omega_i = \mathbf{WordEmbed}(w_i) \oplus \mathbf{CharEmbed}(w_i) \oplus \mathbf{BertEmbed}(w_i).$$

- **WordEmbed** is a simple lookup table.
- **CharEmbed** is implemented using a CNN á la Chiu and Nichols (2016).
- BERT model pre-trained on German text by Chan et al. (2020).

**Encoder:** feed embeddings through a multi-layer bi-directional LSTM with skip-connections and dropout:

$$\mathbf{e} = (\mathbf{e}_i)_{i=0,\dots,L} = \mathbf{BiLSTM}(\omega).$$

( $\mathbf{e}_0$  is the initial state and represents the root pseudo-node.)

**Decoder:** feed embeddings through a single-layer uni-directional LSTM with dropout:

$$\mathbf{d} = (\mathbf{d}_i)_{i=1,\dots,L} = \mathbf{LSTM}((\mathbf{e}_i)_{i=1,\dots,L}).$$

(the initial state of the decoder is the final state of the encoder.)

# Classification tasks: quadratic classifier

Building on Dozat and Manning (2016).

- Model conditional probabilities of  $y_i^{\text{lab}}$ ,  $y_i^{\text{ord}}$ ,  $y_i^{\text{pos}}$  and  $y_i^{\text{morph}}$ .
- Encoder and decoder are shared across tasks.

**Example:** part-of-speech classification.

$$e_i^{\text{pos}} = \text{MLP}_{\text{enc}}^{\text{pos}}(e_i); \quad d_j^{\text{pos}} = \text{MLP}_{\text{dec}}^{\text{pos}}(d_j).$$

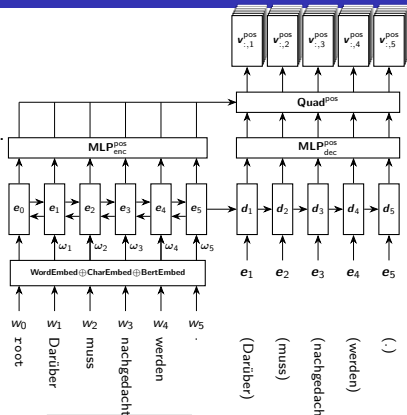
- Obtain class logits  $v_{i,j}^{\text{pos}}$ :

$$v_{i,j}^{\text{pos}} = \text{Quad}^{\text{pos}}(e_i^{\text{pos}}, d_j^{\text{pos}})$$

$$:= e_i^{\text{posT}} \mathbf{U}_{\text{h-d}}^{\text{pos}} d_j^{\text{pos}} + e_i^{\text{posT}} \mathbf{U}_{\text{h-h}}^{\text{pos}} e_i^{\text{pos}} + U_{\text{h}}^{\text{pos}} e_i^{\text{pos}} + d_j^{\text{posT}} \mathbf{U}_{\text{d-d}}^{\text{pos}} d_j^{\text{pos}} + U_{\text{d}}^{\text{pos}} d_j^{\text{pos}} + u_{\text{bias}}^{\text{pos}}.$$

Fixing child  $w_j$ , the vector  $\text{softmax}(v_{i,j}^{\text{pos}})$  can be interpreted as a probability distribution over its parts of speech conditional on having an arc to  $w_i$ :

$$p^{\text{pos}}(c \mid w_i, y_{<j}, w) = \text{softmax}(v_{i,j}^{\text{pos}})_c.$$



# The pointer network: quadratic attention mechanism

Building on Dozat and Manning (2016)'s bi-affine mechanism and drawing from Vinyals et al. (2015).

- Obtain dimension-reduced representations:

$$e_i^{\text{arc}} = \text{MLP}_{\text{enc}_i}^{\text{arc}}(e_i); \quad d_j^{\text{arc}} = \text{MLP}_{\text{dec}_j}^{\text{arc}}(d_j).$$

- Obtain latent features  $v_{i,j}^{\text{arc}}$ :

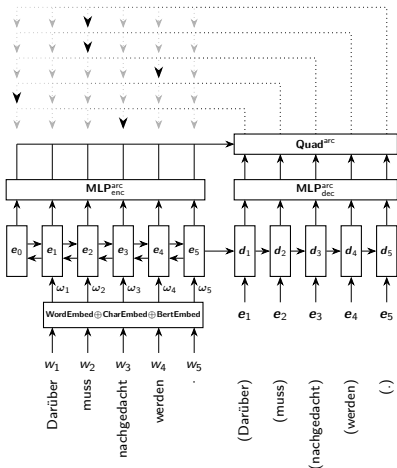
$$\begin{aligned} v_{i,j}^{\text{arc}} &= \text{Quad}^{\text{arc}}(e_i^{\text{arc}}, d_j^{\text{arc}}) \\ &:= e_i^{\text{arcT}} \mathbf{U}_{\text{h-d}}^{\text{arc}} d_j^{\text{arc}} \\ &\quad + e_i^{\text{arcT}} \mathbf{U}_{\text{h-h}}^{\text{arc}} e_i^{\text{arc}} + U_{\text{h}}^{\text{arc}} e_i^{\text{arc}} \\ &\quad + d_j^{\text{arcT}} \mathbf{U}_{\text{d-d}}^{\text{arc}} d_j^{\text{arc}} + U_{\text{d}}^{\text{arc}} d_j^{\text{arc}} + u_{\text{bias}}^{\text{arc}}. \end{aligned}$$

- Obtain attention logits  $s_{i,j}^{\text{arc}}$ :

$$s_{i,j}^{\text{arc}} = \mathbf{u}_{\text{agg}}^{\text{arcT}} \tanh(v_{i,j}^{\text{arc}}).$$

- Fixing child  $w_j$ , the vector  $\text{softmax}(s_{:,j}^{\text{arc}})$  can be interpreted as a probability distribution over potential parents:

$$p^{\text{arc}}(w_i \mid y_{<j}, w) = \text{softmax}(s_{:,j}^{\text{arc}})_i.$$





## Training

We find  $\hat{p}$  using *SGD* with *Nesterov momentum*, finding a suitably low *cross-entropy* between the model  $p$  and the empirical distribution present in the dataset  $(w^n, y^n)_{n=1}^N$ :

$$\begin{aligned} & -\frac{1}{N} \sum_{n=1}^N \log \prod_{i=1}^{L_n} \left\{ p_w(y_i^{n\text{arc}} | y_{<i}^n) p_w(y_i^{n\text{lab}} | y_i^{n\text{arc}}, y_{<i}^n) p_w(y_i^{n\text{ord}} | y_i^{n\text{arc}}, y_{<i}^n) \right. \\ & \quad \left. p_w(y_i^{n\text{pos}} | y_i^{n\text{arc}}, y_{<i}^n) p_w(y_i^{n\text{morph}} | y_i^{n\text{arc}}, y_{<i}^n) \right\} \\ & = \text{loss}^{\text{arc}} + \text{loss}^{\text{lab}} + \text{loss}^{\text{ord}} + \text{loss}^{\text{pos}} + \text{loss}^{\text{morph}}. \end{aligned}$$

## Inference

Given  $w = (w_i)_{i=1}^L$ , estimate  $y = (y_i)_{i=1}^L$  by maximising the estimated conditional probability:

$$\hat{y} = \arg \max_y \left\{ \prod_{i=1}^L \left\{ \hat{p}_w(y_i^{\text{arc}} | y_{<i}) \hat{p}_w(y_i^{\text{lab}} | y_i^{\text{arc}}, y_{<i}) \right. \right. \\ \left. \left. \cdot \hat{p}_w(y_i^{\text{ord}} | y_i^{\text{arc}}, y_{<i}) \hat{p}_w(y_i^{\text{pos}} | y_i^{\text{arc}}, y_{<i}) \hat{p}_w(y_i^{\text{morph}} | y_i^{\text{arc}}, y_{<i}) \right\} \right\}.$$

The feasible region is very large, so the maximisation is approximated via *beam search*.

## Model evaluation

The TIGER treebank is a widely-used corpus of  $\sim 50\,000$  constituent trees. Its main textual basis is the *Frankfurter Rundschau*.

- 97 % of the dataset is usable as training examples, with train/dev/test split of 80/10/10 %.
- PARSEVAL metrics initially proposed by Black et al. (1991):

$$P = \frac{\# \text{ of correct constituents in prediction}}{\# \text{ of total constituents in prediction}}; R = \frac{\# \text{ of correct constituents in prediction}}{\# \text{ of total constituents in reference}}.$$

Model	F1	Disc. F1
Coavoux et al. (2019)	82.7	55.9
Corro (2020)	90.0	62.1
F.-González & G.-Rodríguez (2022)	89.8	71.0
Chen & Komachi (2023)	89.6	70.9
<b>This work</b>	<b>90.59</b>	<b>84.74</b>

*Table:* Comparison of overall F1-score (%) and F1-score measured only on discontinuous constituents (disc. F1). Calculated using disco-dop (Van Cranenburgh et al., 2016) as standard practice. All models configured with BERT.

Model	pos morph (avr)	
Müller et al. (2013)	98.20	98.27
Schnabel & Schütze (2014)	97.50	97.76
Kondratyuk et al. (2018)	98.58	98.97
<b>This work</b>	<b>99.21</b>	<b>99.60</b>

*Table:* Comparison of part of speech (pos) and morphology accuracies (%). Morphology accuracies are the average of accuracies for case, degree, gender, mood, number, person and tense.

## Appendix: Mathematical justification for quadratic classifier

Notation has been simplified for presentational clarity. Single  $c$  is multinoulli and

$$\begin{bmatrix} \mathbf{e}_i \\ \mathbf{d}_j \end{bmatrix} \mid c \sim \mathcal{N} \left( \begin{bmatrix} \boldsymbol{\mu}_c \\ \boldsymbol{\phi}_c \end{bmatrix}, \begin{bmatrix} \mathbf{A}_c & \mathbf{Q}_c^\top \\ \mathbf{Q}_c & \mathbf{B}_c \end{bmatrix}^{-1} \right) \implies \mathbf{d}_j \mid c \sim \mathcal{N}(\boldsymbol{\phi}_c, \mathbf{P}_c^{-1}).$$

The conditional log-probability of  $c$  is the following affine quadratic form:








$$\begin{aligned} \log p(c \mid \mathbf{e}_i, \mathbf{d}_j) &= k_c - \frac{1}{2} \left( (\mathbf{e}_i - \boldsymbol{\mu}_c) + \mathbf{A}_c^{-1} \mathbf{Q}_c^\top (\mathbf{d}_j - \boldsymbol{\phi}_c) \right)^\top \mathbf{A}_c \left( (\mathbf{e}_i - \boldsymbol{\mu}_c) + \mathbf{A}_c^{-1} \mathbf{Q}_c^\top (\mathbf{d}_j - \boldsymbol{\phi}_c) \right) \\ &\quad - \frac{1}{2} (\mathbf{d}_j - \boldsymbol{\phi}_j)^\top \mathbf{P}_c (\mathbf{d}_j - \boldsymbol{\phi}_j) - \log p(\mathbf{e}_i, \mathbf{d}_j) \\ &= - \underbrace{\mathbf{e}_i^\top \mathbf{Q}_c^\top \mathbf{d}_j}_{\text{cth row of } \mathbf{e}_i^\top \mathbf{U}_{h-d} \mathbf{d}_j} - \underbrace{\frac{1}{2} \mathbf{e}_i^\top \mathbf{A}_c \mathbf{e}_i}_{\mathbf{e}_i^\top \mathbf{U}_{h-h} \mathbf{e}_i} - \underbrace{\frac{1}{2} \mathbf{d}_j^\top (\mathbf{P}_c + \mathbf{Q}_c \mathbf{A}_c^{-1} \mathbf{Q}_c^\top) \mathbf{d}_j}_{\mathbf{d}_j^\top \mathbf{U}_{d-d} \mathbf{d}_j} + \underbrace{(\boldsymbol{\mu}_c^\top \mathbf{A}_c + \boldsymbol{\phi}_c^\top \mathbf{Q}_c) \mathbf{e}_i}_{\mathbf{U}_h \mathbf{e}_i} \\ &\quad + \underbrace{(\boldsymbol{\phi}_c^\top \mathbf{Q}_c \mathbf{A}_c^{-1} \mathbf{Q}_c^\top \boldsymbol{\phi}_c + \boldsymbol{\mu}_c^\top \mathbf{Q}_c^\top) \mathbf{d}_j}_{\mathbf{U}_d \mathbf{d}_j} + \underbrace{\left( -\boldsymbol{\mu}_c^\top \mathbf{Q}_c^\top \boldsymbol{\phi}_c - \frac{1}{2} \boldsymbol{\mu}_c^\top \mathbf{A}_c \boldsymbol{\mu}_c - \frac{1}{2} \boldsymbol{\phi}_c^\top (\mathbf{P}_c + \mathbf{Q}_c \mathbf{A}_c^{-1} \mathbf{Q}_c^\top) \boldsymbol{\phi}_c + k_c \right)}_{\text{constant in } c \text{ so throw away}} \\ &= \text{cth row of } \mathbf{e}_i^\top \mathbf{U}_{h-d} \mathbf{d}_j + \mathbf{e}_i^\top \mathbf{U}_{h-h} \mathbf{e}_i + \mathbf{d}_j^\top \mathbf{U}_{d-d} \mathbf{d}_j + \mathbf{U}_h \mathbf{e}_i + \mathbf{U}_d \mathbf{d}_j + \mathbf{u}_{\text{bias}}. \end{aligned}$$

$-\log p(\mathbf{e}_i, \mathbf{d}_j)$   
 constant in  $c$   
 so throw away







**Memory-efficient implementation:** assuming conditional normality:

$$v_{i,j}^c = k^c - \|\mathbf{W}_1^c \mathbf{e}_i + \mathbf{W}_2^c \mathbf{d}_j + \mathbf{w}_3^c\|_2^2.$$

# Bibliography I

- 
- Black, E., Abney, S., Flickinger, D., Gdaniec, C., Grishman, R., Harrison, P., Hindle, D., Ingria, R., Jelinek, F., Klavans, J. L. Et al. (1991). A procedure for quantitatively comparing the syntactic coverage of english grammars. In
- Speech and natural language: Proceedings of a workshop held at pacific grove, california, february 19-22, 1991*
- .
- 
- 
- Chan, B., Möller, T., Pietsch, M. & Soni, T. (2020).
- German BERT (bert-base-german-cased)*
- . Retrieved September 1, 2023, from
- <https://huggingface.co/bert-base-german-cased>
- 
- 
- Chen, Z. & Komachi, M. (2023). Discontinuous combinatory constituency parsing.
- Transactions of the Association for Computational Linguistics*
- , 11, 267–283.
- 
- 
- Chiu, J. P. & Nichols, E. (2016). Named entity recognition with bidirectional lstm-cnns.
- Transactions of the association for computational linguistics*
- , 4, 357–370.
- 
- 
- Coavoux, M., Crabbé, B. & Cohen, S. B. (2019). Unlexicalized transition-based discontinuous constituency parsing.
- Transactions of the Association for Computational Linguistics*
- , 7, 73–89.
- 
- 
- Corro, C. (2020). Span-based discontinuous constituency parsing: A family of exact chart-based algorithms with time complexities from
- $O(n^6)$
- down to
- $O(n^3)$
- . In
- Empirical methods in natural language processing*
- .
- 
- 
- Dozat, T. & Manning, C. D. (2016). Deep biaffine attention for neural dependency parsing. In
- International conference on learning representations*
- .

## Bibliography II

- 
- Eger, S., Gleim, R. & Mehler, A. (2016). Lemmatization and morphological tagging in german and latin: A comparison and a survey of the state-of-the-art. In *Proceedings of the tenth international conference on language resources and evaluation (Irec'16)*.
- 
- Fernández-González, D. & Gómez-Rodríguez, C. (2022). Multitask pointer network for multi-representational parsing. *Knowledge-Based Systems*, 236, 107760.
- 
- Fernández-González, D. & Martins, A. F. (2015). Parsing as reduction. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*.
- 
- Gleim, R., Eger, S., Mehler, A., Uslu, T., Hemati, W., Lücking, A., Henlein, A., Kahlsdorf, S. & Hoenen, A. (2019). A practitioner's view: A survey and comparison of lemmatization and morphological tagging in german and latin. *Journal of Language Modelling*, 7(1), 1–52.
- 
- Kondratyuk, D., Gavenčiak, T., Straka, M. & Hajič, J. (2018). Lemmatag: Jointly tagging and lemmatizing for morphologically-rich languages with brnns. *arXiv preprint arXiv:1808.03703*.
- 
- Müller, T., Schmid, H. & Schütze, H. (2013). Efficient higher-order crfs for morphological tagging. In *Proceedings of the 2013 conference on empirical methods in natural language processing*.



Schnabel, T. & Schütze, H. (2014). FLORS: Fast and simple domain adaptation for part-of-speech tagging. *Transactions of the Association for Computational Linguistics*, 2, 15–26.



Van Cranenburgh, A., Scha, R. & Bod, R. (2016). Data-oriented parsing with discontinuous constituents and function tags. *Journal of Language Modelling*, 4(1), 57–111.



Vinyals, O., Fortunato, M. & Jaitly, N. (2015). Pointer networks. *Advances in neural information processing systems*, 28.